A Appendix

A.1 Proofs

Proof of Lemma 3.2 ($\underline{\mathcal{B}}^*$ is a γ-contraction in ∞-norm). First observe that for any functions f and g,

$$|\max_x f(x) - \max_x g(x)| \leq \max_x |f(x) - g(x)| \tag{14}$$

To see this, suppose $\max_x f(x) > \max_x g(x)$ (the other case is symmetric) and let $\tilde{x} = \arg\max_x f(x)$. Then

$$|\max_x f(x) - \max_x g(x)| = f(\tilde{x}) - \max_x g(x) \leq f(\tilde{x}) - g(\tilde{x}) \leq \max_x |f(x) - g(x)| \tag{15}$$

We also note that (14) implies

$$|\min_x f(x) - \min_x g(x)| \leq \max_x |f(x) - g(x)| \tag{16}$$

since $\min_x f(x) = -\max_x(-f(x))$. Thus for any $Q, Q' : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$,

$$\begin{aligned}
\|\underline{\mathcal{B}}^* Q - \underline{\mathcal{B}}^* Q'\|_\infty &= \sup_{s,a} |\underline{\mathcal{B}}^* Q(s,a) - \underline{\mathcal{B}}^* Q'(s,a)| \\
&= \gamma \sup_{s,a} \left| \min_{s' \in \widehat{T}(s,a)} \max_{a'} Q(s',a') - \min_{s' \in \widehat{T}(s,a)} \max_{a'} Q'(s',a') \right| \\
&\leq \gamma \sup_{s,a} \max_{s' \in \widehat{T}(s,a)} \left| \max_{a'} Q(s',a') - \max_{a'} Q'(s',a') \right| \\
&\leq \gamma \sup_{s',a'} |Q(s',a') - Q'(s',a')| \\
&= \gamma \|Q - Q'\|_\infty
\end{aligned}$$

Hence $\underline{\mathcal{B}}^*$ is indeed a γ-contraction. □

A.2 Extension to stochastic dynamics

Here we outline a possible extension to stochastic dynamics, although we leave experiments with stochastic systems for future work.

First, let us modify the definitions to accommodate stochastic dynamics:

- We introduce safety functions $\mu^\pi(s,a) = \mathbb{E}^\pi[\sum_t \text{Unsafe}(s_t) \,|\, s_0 = s, a_0 = a]$, i.e. Q^π where the cost is the Unsafe indicator and $\gamma = 1$. Note that if an unsafe state is reached, the episode terminates, so the sum is always 0 or 1. In words, $\mu^\pi(s,a)$ is the probability of ever encountering an unsafe state if the agent starts from state s, takes action a, and then follows π thereafter. Similarly, let $\nu^\pi(s) = \mathbb{E}_{a\sim\pi(s)}[\mu^\pi(s,a)]$, analogous to V^π.
- We also define the optimal safety functions $\mu^*(s,a) = \min_\pi \mu^\pi(s,a)$ and $\nu^*(s) = \min_\pi \nu^\pi(s)$.
- A state-action pair (s,a) is p-irrecoverable if $\mu^*(s,a) \geq p$. Otherwise we say that (s,a) is p-safe.
- A state s is p-irrecoverable if $\nu^*(s) \geq p$, and p-safe otherwise.

Our rapid failure assumption must also be extended: There exists a horizon H and threshold q such that if (s,a) is p-irrecoverable, then for any sequence of actions $\{a_t\}_{t=0}^\infty$ with $a_0 = a$, the probability of encountering an unsafe state within H steps is at least q. (Note that necessarily $q \leq p$.)

A.2.1 Analysis

Let s be a p-safe state, and let a and a' be actions where a is p-safe but a' is p-irrecoverable[4]. We want to have $\widetilde{Q}^*(s,a) > \widetilde{Q}^*(s,a')$ so that the greedy policy w.r.t. $\widetilde{Q}^*$, which is an optimal policy for

[4]Note that, as a consequence of the definitions, any action which is p'-safe with $p' < p$ is also p-safe, and similarly any action which is p'-irrecoverable with $p' > p$ is also p-irrecoverable.

$\widetilde{M}$, will only take p-safe actions. Our strategy is to bound $\widetilde{Q}^*(s, a')$ from above and $\widetilde{Q}^*(s, a)$ from below, then choose C to make the desired inequality hold.

We consider a' first, breaking it down into two cases:

- An unsafe state is reached within H steps. Since (s, a') is p-irrecoverable, our assumption implies that an unsafe state is reached within H steps with probability at least q. As calculated in the original submission, the maximum return of a trajectory which is unsafe within H steps is at most $\frac{r_{\max}(1-\gamma^H)-C\gamma^H}{1-\gamma}$. Let us call this constant R_C. If $R_C < 0$, then

$$\mathbb{P}(\text{unsafe within } H \text{ steps}) \cdot (\text{max return} \,|\, \text{unsafe within } H \text{ steps}) \leq qR_C \tag{17}$$

  Otherwise, we can use the fact that any probability is bounded by 1 to obtain

$$\mathbb{P}(\text{unsafe within } H \text{ steps}) \cdot (\text{max return} \,|\, \text{unsafe within } H \text{ steps}) \leq R_C \tag{18}$$

  To satisfy both simultaneously, we can use the bound $\max\{qR_C, R_C\}$.
- The next H states encountered are all safe. This happens with probability less than $1 - q$, and the maximal return is $\frac{r_{\max}}{1-\gamma}$ as usual.

From the reasoning above, we obtain

$$\widetilde{Q}^*(s, a') \leq \mathbb{P}(\text{unsafe within } H \text{ steps}) \cdot (\text{max return} \,|\, \text{unsafe within } H \text{ steps}) + \tag{19}$$

$$\mathbb{P}(\text{safe for } H \text{ steps}) \cdot (\text{max return} \,|\, \text{safe for } H \text{ steps}) \tag{20}$$

$$\leq \max\{qR_C, R_C\} + (1-q)\frac{r_{\max}}{1-\gamma} \tag{21}$$

Now consider a. Since (s, a) is p-safe,

$$\widetilde{Q}^*(s, a) \geq \mathbb{P}(\text{unsafe}) \cdot (\text{min reward} \,|\, \text{unsafe}) + \mathbb{P}(\text{safe}) \cdot (\text{min reward} \,|\, \text{safe}) \tag{22}$$

$$\geq p\left(\frac{-C}{1-\gamma}\right) + (1-p)\frac{r_{\min}}{1-\gamma} \tag{23}$$

$$= \frac{-pC + (1-p)r_{\min}}{1-\gamma} \tag{24}$$

Note that the second step assumes $C \geq 0$. (We will enforce this constraint when choosing C.)

To ensure $\widetilde{Q}^*(s, a) > \widetilde{Q}^*(s, a')$, it suffices to choose C so that the following inequalities hold simultaneously:

$$\frac{-pC + (1-p)r_{\min}}{1-\gamma} > qR_C + (1-q)\frac{r_{\max}}{1-\gamma} \tag{25}$$

$$\frac{-pC + (1-p)r_{\min}}{1-\gamma} > R_C + (1-q)\frac{r_{\max}}{1-\gamma} \tag{26}$$

Multiplying both sides of (25) by $1 - \gamma$ gives the equivalent

$$-pC + (1-p)r_{\min} > qr_{\max}(1-\gamma^H) - qC\gamma^H + (1-q)r_{\max} \tag{27}$$

Rearranging, we need

$$C > \frac{r_{\max}(1-q\gamma^H) - (1-p)r_{\min}}{q\gamma^H - p} =: \alpha_1 \tag{28}$$

Similarly, multiplying both sides of (26) by $1 - \gamma$ gives the equivalent

$$-pC + (1-p)r_{\min} > r_{\max}(1-\gamma^H) - C\gamma^H + (1-q)r_{\max} \tag{29}$$

Rearranging, we need

$$C > \frac{r_{\max}(2-q-\gamma^H) - (1-p)r_{\min}}{\gamma^H - p} =: \alpha_2 \tag{30}$$

All things considered, the inequality $\widetilde{Q}^*(s, a) > \widetilde{Q}^*(s, a')$ holds if we set

$$C > \max\{\alpha_1, \alpha_2, 0\} \tag{31}$$

A.3 Implementation details and hyperparameters

In this appendix we provide additional details regarding the algorithmic implementation, including hyperparameter selection.

Here are some additional details regarding the (S)MBPO implementation:

- All neural networks are implemented in PyTorch [Paszke et al., 2019] and optimized using the Adam optimizer [Kingma and Ba, 2014] and batch size 256.
- The dynamics models use a branched architecture, where a shared trunk computes an intermediate value $z = h_{\theta_1}([s, a])$ which is then passed to branches $\mu_{\theta_2}(z)$ and $\sigma_{\theta_3}(z)$. All three networks are implemented as multi-layer perceptrons (MLPs) with ReLU activation and 200 hidden width. The h_θ network has 3 layers (with ReLU on the final layer too), while μ_{θ_2} and σ_{θ_3} each have one hidden layer (no ReLU on final layer).
- Every 250 environment steps, we update the dynamics models, taking 2000 updates of the Adam optimizer.
- The networks for the Q functions and policies all have two hidden layers of width 256.
- We use a learning rate of 3e-4 for the Q function, 1e-4 for the policy, and 1e-3 for the model.
- Following Fujimoto et al. [2018], we store two copies of the weights for Q (and $\bar{Q}$), trained the same way but with different initializations. When computing the target $\bar{Q}$ in equation (10) and when computing Q in equation (13), we take the minimum of the two copies' predictions. When computing the Q in equation (10), we compute the loss for both copies of the weights and add the two losses.
- When sampling batches of data from $\mathcal{D} \cup \widehat{\mathcal{D}}$, we take 10% of the samples from $\mathcal{D}$ and the remainder from $\widehat{\mathcal{D}}$.

The model-free algorithms have their own hyperparameters, but all share γ_{safe} and ϵ_{safe}. Following Thananjeyan et al. [2020], we tune γ_{safe} and ϵ_{safe} for recovery RL first, then hold those fixed for all algorithms and tune any remaining algorithm-specific hyperparameters. All these hyperparameters are given in the tables below:

Name	Which algorithm(s)?	Choices	hopper	cheetah	ant	humanoid
γ_{safe}	all	0.5, 0.6, 0.7	0.6	0.5	0.6	0.6
ϵ_{safe}	all	0.2, 0.3, 0.4	0.3	0.2	0.2	0.4
ν	LR	1, 10, 100, 1000	1000	1000	1	1
ν	SQRL	1, 10, 100, 1000	1	1000	10	1
λ	RCPO	1, 10, 100, 1000	10	10	1	10

We run our experiments using a combination of NVIDIA GeForce GTX 1080 Ti, TITAN Xp, and TITAN RTX GPUs from our internal cluster. A single run of (S)MBPO takes as long as 72 hours on a single GPU.