Bridging Safety and Performance in Autonomous Systems using Offline Reinforcement Learning

Mumuksh Tayal [*1] **Manan Tayal** [*1] **Ravi Prakash** [1]

Abstract

Offline reinforcement learning (RL) offers a promising framework for deploying autonomous systems in safety-critical settings without the risks of online exploration. However, learning policies that simultaneously achieve high performance and strong safety guarantees from fixed datasets remains a fundamental challenge. Many existing safe offline RL approaches typically rely on soft constraint formulations, which may permit safety violations and are sensitive to distributional shift. In contrast, formal methods such as Hamilton–Jacobi (HJ) reachability and Control Barrier Functions (CBFs) provide rigorous safety guarantees, but often yield overly conservative solutions, often neglecting performance. In this work, we bridge this gap by formulating safe offline RL as a state-constrained optimal control problem, where safety is enforced through hard state constraints and performance is captured via a reward function. The resulting value function satisfies a Hamilton–Jacobi–Bellman (HJB) equation, which we approximate using offline RL on fixed datasets. This formulation enables principled integration of safety guarantees with data-driven policy optimization. Empirically, across safety-critical benchmarks including boat navigation and Safety-Gymnasium tasks, our approach achieves competitive returns while exhibiting near-zero constraint violations, demonstrating a favorable balance between safety and performance in the offline setting.

1. Introduction

Autonomous systems are increasingly deployed in safety-critical domains such as self-driving vehicles (Kiran et al., 2021), robotics, aerospace, and healthcare (Gottesman et al., 2019). In these settings, control design must balance two competing objectives: achieving high task performance while ensuring strict safety. These goals are inherently in tension, as aggressively optimizing performance can increase the risk of unsafe behavior.

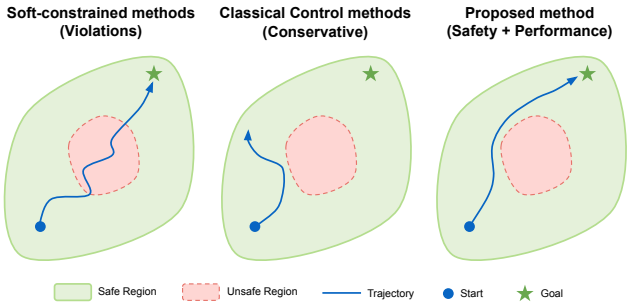

Figure 1. **Safety–performance trade-off in offline RL:** prior methods violate constraints or are overly conservative; Our method achieves safety with high-performance.

Reinforcement learning (RL) (Sutton et al., 1998) provides a powerful framework for learning control policies from data, but standard online exploration is often infeasible in safety-critical settings. Constrained RL methods incorporate safety via Lagrangian penalties or shielding mechanisms (Achiam et al., 2017; Altman, 2021; Alshiekh et al., 2018; Zhao et al., 2023), yet they typically rely on extensive online interaction. This limitation has motivated growing interest in offline RL, where policies are learned solely from pre-collected datasets (Kumar et al., 2020; Levine et al., 2020).

Despite this promise, achieving reliable safety in offline RL remains difficult. Many existing safe offline RL methods enforce safety through expected cumulative costs or penalties (Xu et al., 2022; Stooke et al., 2020), effectively treating safety as a soft constraint. Such formulations allow non-zero violation risk and are therefore unsuitable for applications that require state-wise safety guarantees, where even a single violation is unacceptable. Moreover, jointly optimizing performance and safety from static datasets often leads to unstable learning dynamics or overly conservative policies, particularly when unsafe transitions are rare or entirely absent from the data (Lee et al., 2022). Complementary efforts have adapted classical control tools to offline set-

[1]Indian Institute of Science (IISc), Bangalore, India. Correspondence to: Manan Tayal <tayalmanan28@gmail.com>.

Published at the Second Workshop on Agents in the Wild: Safety, Security, and Beyond (AIWILD) at ICML 2026. Copyright 2026 by the author(s).

tings, including learning Control Barrier Functions (CBFs) and Hamilton–Jacobi (HJ) reachability-based policies from demonstrations (Robey et al., 2020; Castañeda et al., 2023; Zheng et al., 2024; Tayal et al., 2026). While these approaches provide stronger safety guarantees, they typically operate as safety filters and often sacrifice performance.

A principled alternative is to formulate the problem as a *state-constrained optimal control problem* (SC-OCP), where safety is enforced as a hard state constraint and performance is captured via a reward functional. Despite its strong theoretical foundations, solving SC-OCPs in practice is difficult: the existence and characterization of optimal solutions typically rely on stringent controllability conditions (Soner, 1986). To mitigate these difficulties, (Altarovici et al., 2013) introduced an epigraph-based reformulation that characterizes the SC-OCP value function by computing its epigraph through dynamic programming, leading to a Hamilton–Jacobi–Bellman partial differential equation (HJB-PDE). The original value function and the associated optimal policy can then be recovered from this epigraph representation. Nevertheless, solving HJB-PDEs fundamentally limited by the curse of dimensionality, rendering them impractical for high-dimensional systems when using conventional numerical solvers (Mitchell, 2004; Wang et al., 2024). The epigraph reformulation further augments the state space, compounding the computational burden and limiting applicability to low-dimensional settings. Although various approximation and acceleration strategies have been proposed, they often require restrictive assumptions on system structure or dynamics (Chow et al., 2017). Consequently, computing solutions to the resulting HJB equations for general nonlinear, high-dimensional control systems remains an open and challenging problem.

In this work, we bridge the gap between rigorous state-constrained control and scalable offline RL by deriving a data-driven Bellman-style recursion for an *epigraph feasibility value function*. This function captures the maximum achievable performance that remains compatible with future safety, enabling feasibility-aware value learning directly from offline data without solving high-dimensional HJB equations. To extract policies, we leverage Flow Matching as an efficient generative mechanism, guiding action synthesis toward regions that are both safe and high-performing within the support of the dataset.

We evaluate the proposed approach across a range of safety-critical benchmarks, including navigation tasks and high-dimensional Safety Gymnasium environments (Ji et al., 2023). Our results demonstrate competitive returns with near-zero empirical safety violations, outperforming both penalty-based and safety-filtering baselines.

The main contributions of this work are:

- We formulate safe offline RL as a state-constrained optimal control problem, providing a principled framework for jointly optimizing safety and performance.
- We derive a bounded Bellman-style recursion for an epigraph value function, to solve for the resultant HJB-PDE, thereby enabling feasibility-aware learning directly from offline data.
- We incorporate this formulation into a generative policy framework using Flow Matching, enabling efficient synthesis of safe and high-performing actions in continuous spaces.
- We demonstrate strong empirical performance across safety-critical benchmarks, achieving near-zero violations while maintaining competitive returns.

2. Background and Problem Setup

We study safe offline reinforcement learning in environments with hard state constraints. The environment is modelled as a state-constrained Markov Decision Process (SC-MDP), defined by the tuple $\mathcal{M} = (\mathcal{X}, \mathcal{A}, f, r, \ell, \gamma)$, where $\mathcal{X}$ and $\mathcal{A}$ denote the state and action spaces, $x_{t+1} = f(x_t, a_t)$ denotes a unknown, deterministic transition function, $r : \mathcal{X} \to \mathbb{R}$ is the reward function, $\ell : \mathcal{X} \to \mathbb{R}$ is an instantaneous state-based safety function, and $\gamma \in (0, 1)$ is the discount factor. We define the *failure set* $\mathcal{F} := \{x \in \mathcal{X} \mid \ell(x) < 0\}$, which represents unsafe states that must be avoided at all times (e.g., collisions or constraint violations). A trajectory is considered safe if it never enters $\mathcal{F}$. We assume access to an offline dataset $\mathcal{D} = \{(x_t, a_t, r_t, \ell_t, x_{t+1})\}$, collected by an unknown behavior policy, with no further interaction with the environment permitted. A policy $\pi(a \mid x)$ induces trajectories $\Gamma = (x_0, a_0, x_1, \dots)$ through the transition dynamics.

2.1. State-Constrained Optimal Control Problem

Given an initial state x, the objective is to compute the maximum achievable discounted return subject to strict state safety at all future time steps. This requirement can be formalized through the following *state-constrained value function*:

$$V(x) \;=\; \sup_{\pi} \big[\sum_{k=0}^{\infty} \gamma^k r(x_k) \,\big|\, x_0 = x \big], \;\; \text{s.t. } \; x_t \notin \mathcal{F}, \; \forall t \geq 0. \tag{1}$$

Unlike formulations based on expected cumulative penalties, equation (1) encodes a *hard safety requirement*, only policies that admit trajectories remaining entirely outside the failure set are considered feasible. This formulation directly captures safety-critical requirements where even a single violation is unacceptable.

However, directly optimizing equation (1) in the offline setting is challenging. Feasibility depends on long-horizon future behavior, which cannot be verified using local penalties or one-step constraints. Moreover, offline datasets often underrepresent unsafe transitions, making it difficult to infer whether a given state admits any safe continuation. As a result, standard offline RL objectives, which optimize expected return, are ill-suited for enforcing state-wise safety constraints.

2.2. Epigraph Reformulation

The state-constrained value function in equation (1) jointly optimizes performance and safety, which makes it difficult to reason about directly in the offline setting. To expose the structure of this problem, we adopt an epigraph-based reformulation that expresses the original objective as a two-stage optimization.

In the first stage, the state-constrained value function is characterized as the largest performance threshold that is feasible under the safety constraint:

$$V(x) = \sup\{z \in \mathbb{R} \mid \hat{V}(x, z) \geq 0\}. \tag{2}$$

This formulation makes explicit that $V(x)$ represents the maximum achievable return level that is compatible with safety starting from state x.

The second stage defines an auxiliary value function $\hat{V}(x, z)$, which evaluates whether a given performance threshold z is feasible. Specifically, $\hat{V}(x, z)$ is defined as

$$\hat{V}(x_t, z_t) = \sup_{\pi} \min \Big\{ \sum_{k=t}^{\infty} \gamma^{k-t} r(x_k) - z_t,\ \min_{s \geq t} \gamma^{s-t} \ell(x_s) \Big\}. \tag{3}$$

A formal derivation of the epigraph reformulation is provided in Appendix A.1. Discounting is applied to the safety term solely to enable a Bellman-style discrete-time recursion (Equation (5)); since $\gamma \in (0, 1)$ preserves the sign of $\ell(x)$, this does not alter the hard state constraint and serves only as a technical device for dynamic programming (see Appendix A.1). This epigraph feasibility value function $\hat{V}(x, z)$ is non-negative if and only if there exists a policy that both achieves discounted return at least z and remains outside the failure set at all future time steps. As a result, equation (2) selects the largest performance threshold for which the safety constraint is feasible.

The scalar variable z therefore plays the role of a performance envelope. Values of z smaller than the optimal threshold correspond to conservative but safe behavior, while values larger than this threshold are infeasible and necessarily lead to safety violations.

For discounted infinite-horizon problems, the auxiliary epigraph variable admits the discrete-time update

$$z_{t+1} = \frac{z_t - r(x_t)}{\gamma}, \tag{4}$$

which ensures that the augmented state (x_t, z_t) evolves as a Markov process. This interpretation enables learning the feasibility structure of $V(x)$ from offline data using value-based methods, without directly enforcing constraints during policy optimization.

3. Methodology

In Section 2 we showed that the state-constrained value function $V(x)$ may be characterized via the feasibility epigraph value function $\hat{V}(x, z)$ through the two-stage formulation (Equations (2),(3)). This section describes a practical, data-driven procedure to learn a dataset-consistent policy that is both safe and performant. We first derive the bounded recursive bellman equation for learning epigraph feasibility value function from offline data and then using it to synthesize a policy. Implementation-level details are summarized in Algorithm 1 & Algorithm 2 and illustrated in Fig. 2.

3.1. Recursive Characterization

We begin by characterizing the structure of the feasibility epigraph value function. In the online setting, a Bellman-style recursion alone suffices for learning $\hat{V}$ (So & Fan, 2023). However, in the offline setting the recursion is ill-conditioned: because the Bellman update does not explicitly enforce dependence on the epigraph variable z, function approximators can become desensitized to z, causing $\hat{V}$ to collapse toward the safety value V_s (see Appendix A.4). To address this, we state a *bounded* recursion that couples the Bellman equation with a decomposition upper bound derived from the epigraph definition.

Theorem 3.1 (Bounded Epigraph Value Recursion)**.** *Consider a deterministic SC-MDP with bounded reward and discount factor $\gamma \in (0, 1)$. Given a transition $(x_t, a_t, r_t, \ell_t, x_{t+1})$ and the epigraph update $z_{t+1} = (z_t - r(x_t))/\gamma$, define the optimal reward value $V_r(x_t) = \sup_{\pi} \sum_{k=t}^{\infty} \gamma^{k-t} r(x_k)$ and the optimal safety value $V_s(x_t) = \sup_{\pi} \min_{s \geq t} \gamma^{s-t} \ell(x_s)$. Then the epigraph value function $\hat{V}(x_t, z_t)$ satisfies:*

(i) Bellman recursion.

$$\hat{V}(x_t, z_t) \ = \ \min \big\{ \ell(x_t),\ \gamma \hat{V}(x_{t+1}, z_{t+1}) \big\}. \tag{5}$$

(ii) Decomposition upper bound.

$$\hat{V}(x_t, z_t) \ \leq \ \min \big\{ V_r(x_t) - z_t,\ V_s(x_t) \big\}. \tag{6}$$

A formal proof is provided in Appendix A.2. Part (i) follows from decomposing the discounted return and safety terms in equation (3), and convergence can be established using arguments similar to those for safety value functions in Hamilton–Jacobi reachability (Fisac et al., 2019; So & Fan, 2023). Part (ii) follows from the sup-min inequality applied to the epigraph definition (3).

Why both parts are necessary. Part (i) alone is insufficient because the Bellman update for $\hat{V}(x, z)$ is structurally identical to that of a pure safety value function:

$$\begin{aligned} V_s(x_t) &= \min\{\ell(x_t),\ \gamma V_s(x_{t+1})\} \\ &\updownarrow \\ \hat{V}(x_t, z_t) &= \min\{\ell(x_t),\ \gamma \hat{V}(x_{t+1}, z_{t+1})\}. \end{aligned} \tag{7}$$

As a result, under limited offline data, a function approximator can minimize training error by learning a degenerate solution that ignores the epigraph variable and collapses $\hat{V}(x, z)$ to a state-only safety value $V_s(x)$. This collapse arises because z influences the recursion only through a shifted argument, making its effect weak and difficult to identify from data (see Appendix A.4).

However, this behavior is incorrect, in safe regions the value function should depend explicitly on the performance term $V_r(x) - z$. Part (ii) prevents this collapse by introducing an explicit linear dependence on z, forcing the model to retain sensitivity to the epigraph variable. Together, the two parts ensure that $\hat{V}(x, z)$ captures both safety and performance, yielding a well-posed fixed-point suitable for offline learning (a detailed discussion is in Appendix A.3).

Avoiding OOD actions. The implicit supremum over policies in part (i) (Theorem 3.1) can produce optimistic values for actions not in the dataset. We therefore adopt expectile regression (Kostrikov et al., 2022): an action-conditional $\hat{Q}(x, z, a)$ is fitted on dataset transitions and then distilled into $\hat{V}(x, z)$ via the asymmetric loss $\mathcal{L}^\tau(u) = |\tau - \mathbf{1}(u < 0)|\, u^2$ with $\tau \in (0.5, 1)$, which biases toward the upper envelope of dataset-supported values without querying unseen actions.

3.2. Practical Loss Functions

Learning proceeds by alternating fitting an action-value model, distilling expectile values, estimating the separate reward and safety envelopes, and enforcing the decomposition constraint. All expectations are empirical averages over minibatches drawn from $\mathcal{D}$ together with z values sampled from a dataset-supported interval $[z_{\min}, z_{\max}]$. The feasibility Q-target follows from equation (5),

$$y_Q(x, z, a, x', z') \;=\; \min\{\ell(x),\ \gamma \hat{V}_\theta(x', z')\}, \tag{8}$$

the corresponding squared loss for which is

$$\mathcal{L}_{\hat{Q}}(\theta') = \mathbb{E}\big[(y_Q(x, z, a, x', z') - \hat{Q}_{\theta'}(x, z, a))^2\big]. \tag{9}$$

Expectile distillation to the state-value is performed by

$$\mathcal{L}_{\hat{V}}(\theta) = \mathbb{E}\big[\mathcal{L}^\tau(\hat{Q}_{\theta'}(x, z, a) - \hat{V}_\theta(x, z))\big], \tag{10}$$

with τ chosen to bias toward the upper data-supported envelope. To enforce the decomposition upper bound (Theorem 3.1, part (ii)), we independently learn reward and safety value functions $V_r(x)$ and $V_s(x)$ using standard IQL-style losses (details in Appendix A.5). The decomposition regularizer is then implemented as the one-sided penalty

$$\begin{aligned} \mathcal{L}_{reg}(\theta) \;=\; \mathbb{E}\Big[\max\Big(0,\ \hat{V}_\theta(x, z) \\ - \min\big\{V_r^\phi(x) - z,\ V_s^\psi(x)\big\}\Big)\Big], \end{aligned} \tag{11}$$

reduction of this loss would ensure satisfaction of equation (6) and the full objective combines the terms:

$$\mathcal{L}(\theta) \;= \mathcal{L}_{\hat{V}}(\theta) + \lambda \cdot \mathcal{L}_{reg}(\theta) \tag{12}$$

where the λ controls the trade-offs between bellman regression and the decomposition regularizer.

The outcome of this learning procedure is a dataset-consistent feasibility value function $\hat{V}(x, z)$ together with reward and safety envelopes $V_r(x)$ and $V_s(x)$. The complete learning procedure is summarized in Algorithm 1, and its interaction between the different components is illustrated in Fig. 2.

3.3. Policy Synthesis

Given the learned epigraph feasibility value function $\hat{V}_\theta(x, z)$, we now describe how to synthesize a policy that improves performance while respecting the feasibility constraints. For each state x, feasibility induces a maximal admissible performance threshold

$$z^*(x) = \sup\{z \in \mathbb{R} \mid \hat{V}_\theta(x, z) \geq 0\}, \tag{13}$$

which is obtained via binary search over z. Although strict monotonicity is not enforced architecturally, we find that the learned $\hat{V}_\theta(x, z)$ exhibits sufficient empirical monotonicity for stable bisection when combined with interval clamping.

Policy improvement seeks actions with high feasible value while staying near the offline data distribution. While recent offline RL methods often use weighted regression with rejection sampling (Liu et al., 2025; Wang et al., 2023; Zheng et al., 2024), such sampling increases inference cost. We therefore follow an approach similar to Flow

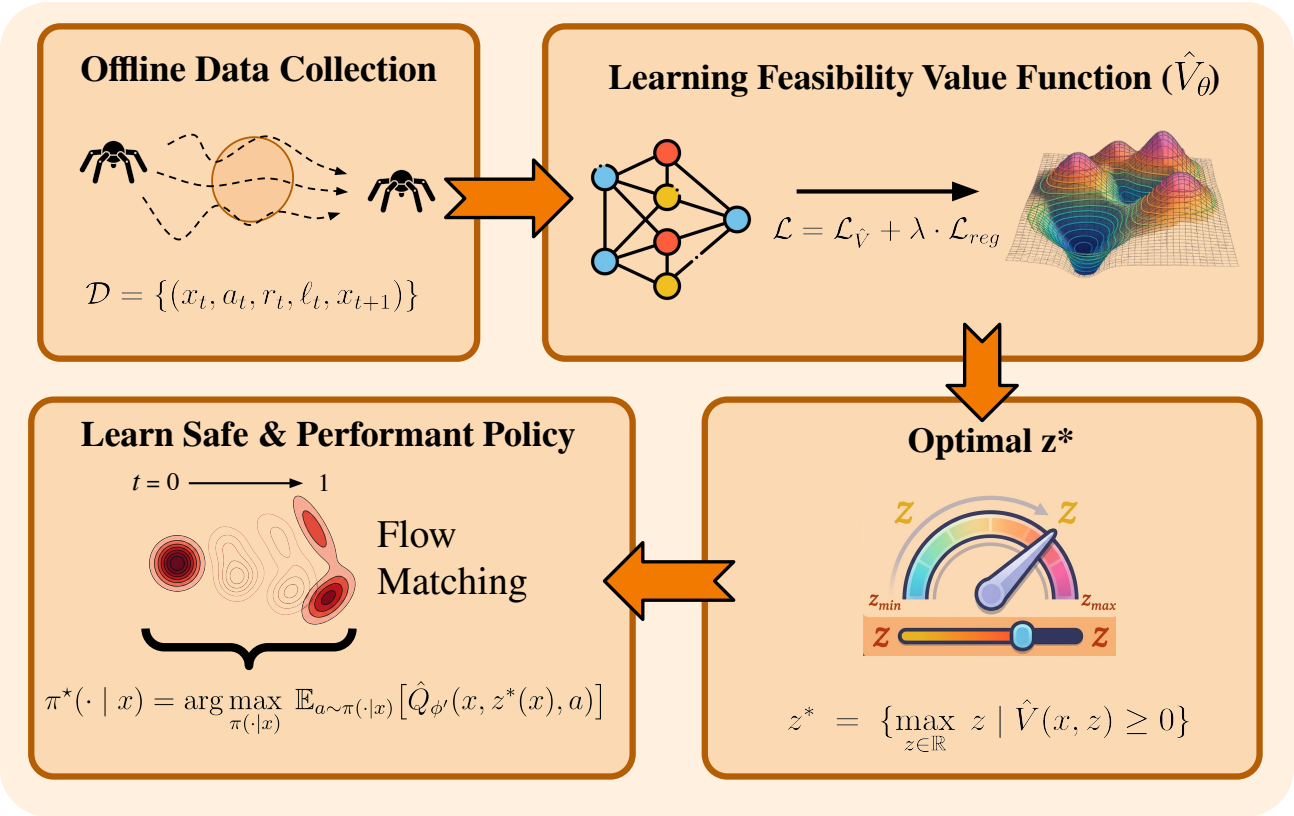

Figure 2. **Framework Overview.** We propose a framework to solve the SC-OCP problem by using offline dataset. We learn $\hat{V}$, which if satisfies to stay positive, makes sure that the system stays safe while maximizing the reward. We then learn a policy which optimizes for $\hat{V}$ at all time steps to find the most optimal path that the offline data supports.

Q-Learning (Park et al., 2025) and use a dataset behavior policy (π_β) based flow-matching model as a *flow behavior teacher* (μ_ζ), which is distilled into a one-step actor (μ_ω) that directly optimizes

$$\pi^\star(\cdot \mid x) = \arg\max_{\pi(\cdot|x)} \mathbb{E}_{a\sim\pi(\cdot|x)}\big[\hat{Q}_{\theta'}(x, z^*(x), a)\big] \quad \text{s.t. } Div\big(\pi(\cdot \mid x) \,||\, \pi_\beta(\cdot \mid x)\big) \leq \epsilon, \tag{14}$$

where μ_ω is the desired $\pi^\star$, Div depicts divergence, $\pi_\beta(a \mid x)$ denote the behavior policy and ϵ is a small positive constant.

Learning the Policy. To optimize for $\hat{Q}_{\theta'}(x, z^*(x), a)$ while staying close to the offline data distribution, a *flow behavior teacher* $\mu_\zeta(x, \eta)$ is first trained by conditional flow matching, where x is the state and $\eta \sim \mathcal{N}(0, I)$ is a latent noise variable. The teacher is parameterized by a time-dependent velocity field $v_\zeta(x, \eta, t)$ and is trained with the regression objective

$$\mathcal{L}_{\text{flow}}(\zeta) = \mathbb{E}_{\substack{(x,a)\sim\mathcal{D},\\ \eta\sim\mathcal{N}(0,I),\\ t\sim\mathcal{U}([0,1])}} \left[\|v_\zeta(x, \eta_t, t) - (a - \eta)\|_2^2\right], \tag{15}$$

where $\eta_t = (1 - t)\eta + ta$. After training, integrating v_ζ from $t = 0$ to $t = 1$ maps η to an action sampled from the data-supported behavior policy. We then train a one-step student actor $\mu_\omega(x, \eta)$ that directly predicts an action from (x, η) without iterative integration. The student is distilled from the teacher through

$$\mathcal{L}_{\text{distill}}(\omega) = \mathbb{E}_{(x,\eta)\sim\mathcal{D}\times\mathcal{N}(0,I)}\left[\|\mu_\omega(x, \eta) - \mu_\zeta(x, \eta)\|_2^2\right], \tag{16}$$

which regularizes the actor toward the support of the offline dataset and reduces extrapolation error. Finally, we train the deployment policy with

$$\mathcal{L}_{\text{actor}}(\omega) = \mathbb{E}_{x\sim\mathcal{D},\, \eta\sim\mathcal{N}(0,I)}\big[-\hat{Q}_{\theta'}(x, z^*(x), \mu_\omega(x, \eta))\big] + \alpha\mathcal{L}_{\text{distill}}(\omega), \tag{17}$$

where α balances Q-maximization and behavior regularization. Since μ_ω is a one-step policy, it is the only component queried at deployment, giving a single forward pass per action.

4. Experiments

Evaluation Metrics: We would like to answer two questions: (i) how our method's safety rate compares against state-of-the-art safe offline RL baselines, and (ii) how safety

compliance affects task performance, measured by the cumulative reward $R(x) = \sum_{k=0}^{\infty} \gamma^k r(x_k)$. Accordingly, we evaluate methods along two axes: *safety/cost*, measured by the total number of safety violations before episode termination, and *performance*, measured by cumulative episode reward. This allows us to assess the safety–performance trade-off across different offline RL approaches.

Baselines: We evaluate against two classes of safe offline RL methods that reflect different trade-offs between performance and safety. The first class focuses on maximizing return while incorporating safety constraints, typically through Lagrangian or regularization-based formulations. This includes **BEAR-Lag** (a Lagrangian variant of (Kumar et al., 2019)), **BCQ-Lag** (a Lagrangian extension of (Fujimoto et al., 2019)), as well as constraint-aware offline RL methods such as **COptiDICE** (Lee et al., 2022), **CPQ** (Xu et al., 2022), and **C2IQL** (Liu et al., 2025). The second class emphasizes strict safety, often at the expense of reward maximization, and includes methods that prioritize absolute safety. This category comprises of offline HJ Reachability and Control Barrier Function (CBF)-based approaches such as **FISOR** (Zheng et al., 2024), **NCBF** (Robey et al., 2020) and **V-OCBF** (Tayal et al., 2026).

4.1. Experimental Case Studies

For experiments, we test and evaluate our framework against the baselines on a varied range of environments. Below we list all these environments that we use:

- **Safe Boat Navigation:** In our first experiment, we examine a 2-dimensional collision avoidance problem involving a boat in a river with drift velocity that varies with the y-coordinate of the boat. The objective is to ensure safety by avoiding the static obstacles while navigating through the river with varying drift. Further details on the system dynamics, state space bounds, and experimental setup are provided in Appendix B.1.
- **Safety Gymnasium:** We also evaluate our framework on Safety Gymnasium environments (Ji et al., 2023) which include Safe Velocity and Safe Navigation Gymnasium environments. Specifically, we use high-dimensional MuJoCo tasks like Hopper, Half Cheetah, Swimmer, Walker2D and Ant for Safe Velocity task, and Car Navigation task with 3 navigation environment types: Goal, Push, Button.

For all standard Safety Gymnasium environments, we retain the original reward and safety-violation metrics from Safety-Gymnasium (Ji et al., 2023) and use standard DSRL dataset for safe offline RL (Liu et al., 2024). To evaluate our method against baselines on all the environments, we fix 500 randomly sampled safe initial states for each environment and perform rollouts over 5 different seeds.

4.2. Results

We evaluate our framework against a comprehensive set of offline RL baselines across all environments described above. Figure 3 summarizes the results, illustrating how different methods navigate the trade-off between cumulative reward and safety constraint satisfaction. Existing safe offline RL approaches vary widely in their ability to balance these objectives. Lagrangian-based methods such as **BEAR-Lag** and **BCQ-Lag** frequently favor reward maximization, resulting in elevated safety violations due to weak constraint enforcement. Other approaches, including **COptiDICE**, **CPQ**, and **C2IQL**, exhibit inconsistent behavior across tasks, alternating between overly conservative policies that sacrifice performance and policies that fail to maintain reliable safety rates. In contrast, strong safety-constrained methods such as **FISOR**, **NCBF**, and **V-OCBF** achieve strong safety performances across environments, but typically doing so by adopting conservative policies that limit achievable reward.

On the other hand, our framework consistently attains high safety rates while maintaining strong performance across all evaluated domains, ranging from low-dimensional Boat Navigation to high-dimensional MuJoCo Safe Velocity and Safe Car Navigation tasks. Unlike prior methods that trade reward for safety or vice versa, our framework reliably operates in the desirable regime where both objectives are jointly optimized. By leveraging the epigraph feasibility value function, our approach anticipates future safety violations and allocates performance budgets accordingly, avoiding both excessive conservatism and unsafe reward-seeking. These results demonstrate that our framework offers a more stable and effective solution for safety-critical offline reinforcement learning than existing constraint-based or filtering-based baselines.

4.3. Ablation Studies

To better understand the design choices and robustness, we conduct a series of ablation studies that isolate the effect of key modeling and inference components.

Effect of Regularizer Parameter (λ). We study how the decomposition regularizer (term $\mathcal{L}_{reg}$ in equation (12)) affects the learned feasibility epigraph value $\hat{V}_\theta$ by sweeping $\lambda \in \{0.1, 0.25, 0.5, 1.0\}$. Figure 4 summarizes downstream metrics (mean reward, mean cost).

The results in Figure 4 indicate that across different values of λ, a value which balances safety and reward somewhere in the middle ground should be optimal, the learned value function shouldn't drift away from the primary recursion formulation objective while also strictly complying with the upper bound. Based on these results, we fixate to using $\lambda = 0.25$ as the preferred regularizer constant for our experiments. For more detailed explanation of this experiment,

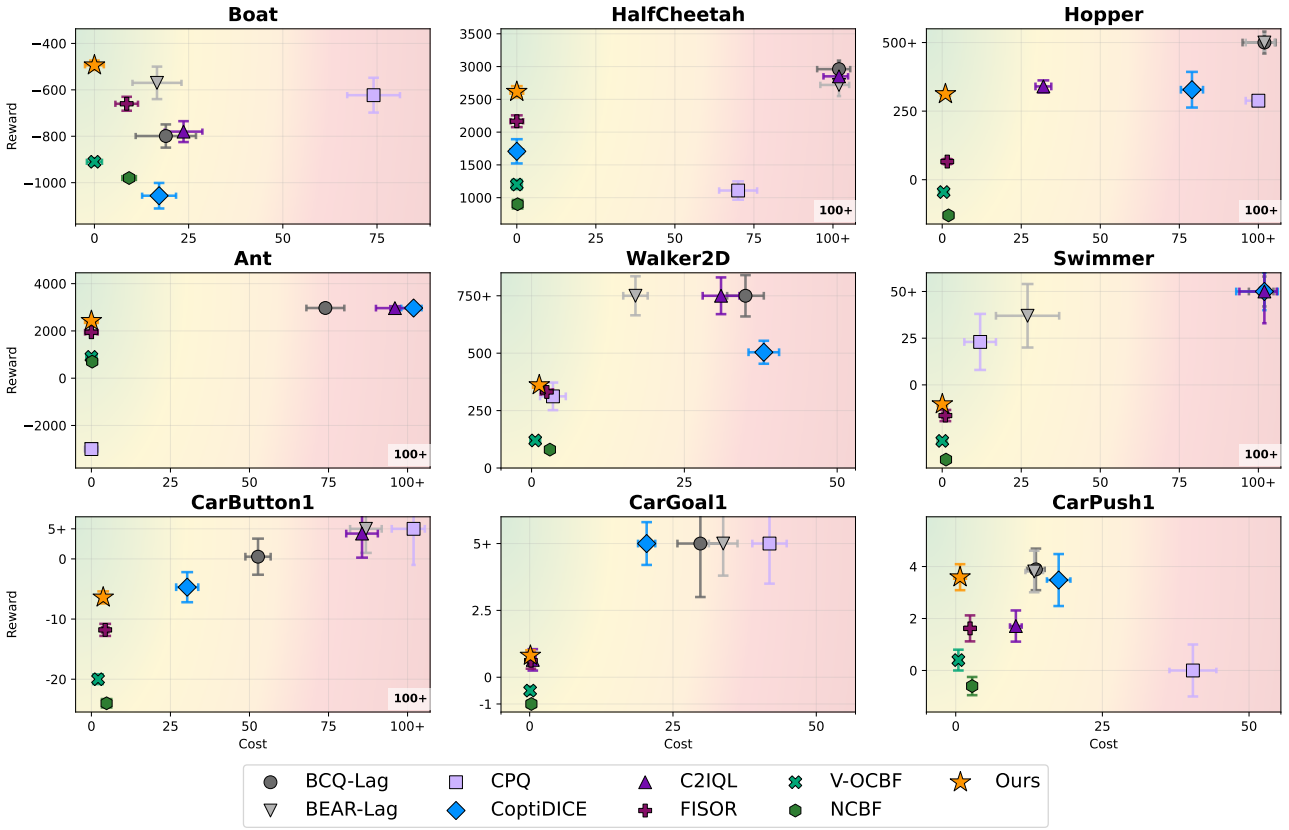

Figure 3. **Evaluation Results:** Results plot for all the environments. Points towards left (←) are more safe than those on right (→). Whereas those towards top (↑) have higher rewards than those towards bottom (↓). Evaluated over 500 episodes and 5 seed values (± standard deviation).

refer Appendix C.3.

Sensitivity to Expectile Parameter (τ). To analyze the sensitivity of the expectile regression parameter used to learn state value functions in Section 3, we perform an ablation study to understand how this expectile parameter τ would affect the reward performance and safety in terms of cost. From the Figure 5, we witness that a higher τ yields higher safety while keeping rewards competitive.

As our original objective from Section 2.1 is to comply with the constraint while maximizing the reward, it is essential that we pick a τ so that it achieves highest safety rates. Therefore, based on this ablation study, we prefer $\tau = 0.9$ expectile parameter for all our experiments. A more detailed study on this ablation can be referred to in Appendix C.1.

Computation Time Analysis The plots in Fig. 6 illustrate a variance in computational demands among existing methods when studied over 5 trials, some baselines require extensive training times while others face significant inference time latency. In contrast, our proposed framework naturally maintains a balance, staying on the lower side of both metrics. Overall, this demonstrates that our method does not require extensive compute resources and can be implemented efficiently within a highly reasonable timeframe.

Additional Ablation Studies. We further perform some more ablations to examine (i) the robustness of the learned policy to action perturbations, which probes the stability of epigraph-guided synthesis under execution noise; and (ii) perform a comparison with Diffusion Model based counterpart in place of Flow Matching for our framework to show that our framework is agnostic of any generative model. Together, these studies provide insight into the stability, hyperparameter sensitivity, and practical deployment characteristics of our framework. Detailed quantitative results and visualizations are reported in Appendix C.

5. Conclusion and Future Work

In this work, we proposed an offline reinforcement learning framework that co-optimizes safety and performance by casting state-constrained control as a tractable epigraph-based value synthesis problem. The method learns an epigraph feasibility value function via bounded bellman equation and synthesizes executable continuous-action policies

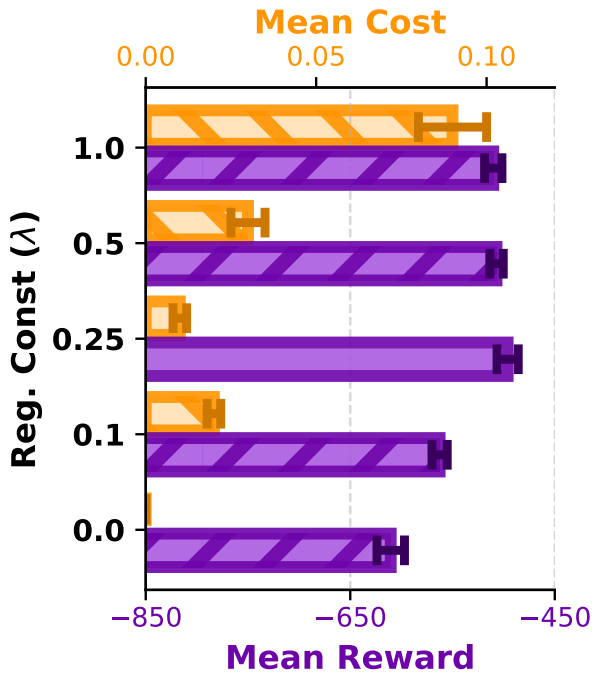

Figure 4. Effect of Regularizer parameter (λ) ($\pm$ standard deviation).

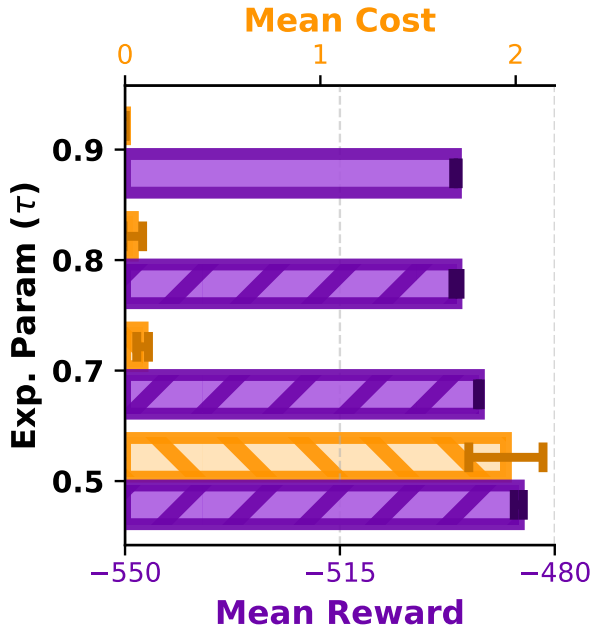

Figure 5. Sensitivity to Expectile Parameter (τ) ($\pm$ standard deviation).

using flow matching, while remaining grounded in the offline dataset. Empirically, the approach achieves consistently low safety violations across high-dimensional control tasks while maintaining competitive performance.

While our framework reliably learns safe and performant policy from offline demonstrations, it is not formally verified, function approximation errors mean guarantees remain empirical. Furthermore, the framework is not explicitly trained to handle adversarial disturbances. To close this gap, future work will explore integration of formal-certification (e.g., Lipschitz-based verification (Tayal et al., 2024) and conformal prediction (Lindemann et al., 2025)). Moreover, we will explore robust extensions of the framework that incorporate adversarial or uncertainty-aware training objectives.

Impact Statement

This work introduces a robust methodology for co-optimizing safety and performance in autonomous systems within the offline reinforcement learning paradigm. By utilizing an epigraph-based reformulation to transform hard state constraints into a tractable optimization problem, our framework enables the extraction of reliable policies from pre-collected, static datasets without the risks associated with online exploration. The integration of weighted Flow Matching ensures that the synthesized controllers remain strictly faithful to the safe regions of the data distribution, effectively mitigating the dangers of over-estimation in safety-critical environments. These advancements provide a scalable path for deploying reliable agents in high-dimensional domains, such as robotics and autonomous mobility, where learning exclusively from demonstrations is a prerequisite for operational trust and public safety.

References

Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 22–31. JMLR.org, 2017.

Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., and Topcu, U. Safe reinforcement learning via shielding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.11797. URL https://ojs.aaai.org/index.php/AAAI/article/view/11797.

Altarovici, A., Bokanowski, O., and Zidani, H. A general hamilton-jacobi framework for non-linear state-constrained control problems. *ESAIM: Control, Optimisation and Calculus of Variations*, 19(2):337–357, 2013.

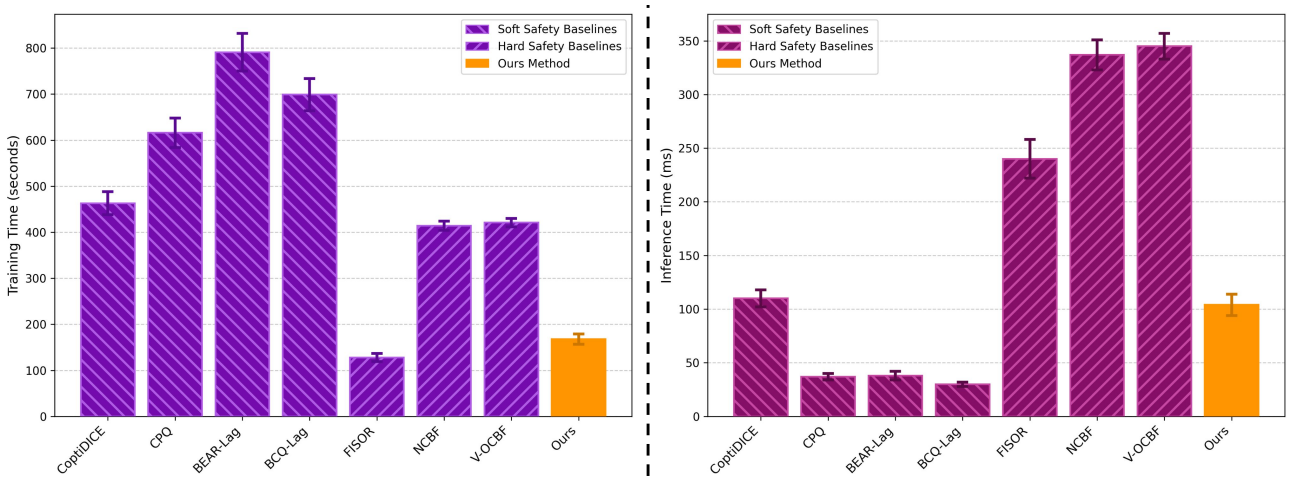

Figure 6. **Computation Time Analysis.** *(Left)* shows training time comparison while *(Right)* shows inference time (± standard deviations) comparison of Our framework against all baselines.

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

Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Castañeda, F., Nishimura, H., McAllister, R. T., Sreenath, K., and Gaidon, A. In-distribution barrier functions: Self-supervised policy filters that avoid out-of-distribution states. In Matni, N., Morari, M., and Pappas, G. J. (eds.), *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, volume 211 of *Proceedings of Machine Learning Research*, pp. 286–299. PMLR, 15–16 Jun 2023. URL https://proceedings.mlr.press/v211/castaneda23a.html.

Chow, Y. T., Darbon, J., Osher, S., and Yin, W. Algorithm for overcoming the curse of dimensionality for time-dependent non-convex hamilton–jacobi equations arising from optimal control and differential games problems. *Journal of Scientific Computing*, 73:617–643, 2017.

Fisac, J. F., Lugovoy, N. F., Rubies-Royo, V., Ghosh, S., and Tomlin, C. J. Bridging hamilton-jacobi safety analysis and reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8550–8556, 2019. doi: 10.1109/ICRA.2019.8794107.

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pp. 2052–2062. PMLR, 2019.

Gottesman, O., Johansson, F., Komorowski, M., Faisal, A., Sontag, D., Doshi-Velez, F., and Celi, L. A. Guidelines for reinforcement learning in healthcare. *Nature medicine*, 25(1):16–18, 2019.

Ji, J., Zhang, B., Zhou, J., Pan, X., Huang, W., Sun, R., Geng, Y., Zhong, Y., Dai, J., and Yang, Y. Safety gymnasium: A unified safe reinforcement learning benchmark. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=WZmlxIuIGR.

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S., and Pérez, P. Deep reinforcement learning for autonomous driving: A survey. *IEEE transactions on intelligent transportation systems*, 23(6): 4909–4926, 2021.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in neural information processing systems*, 32, 2019.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/0d2b2061826a5df3221116a5085a6052-Paper.pdf.

Lee, J., Paduraru, C., Mankowitz, D. J., Heess, N., Precup, D., Kim, K.-E., and Guez, A. COptiDICE: Offline constrained reinforcement learning via stationary distribution correction estimation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=FLA55mBee6Q.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020. URL https://arxiv.org/abs/2005.01643.

Lindemann, L., Zhao, Y., Yu, X., Pappas, G. J., and Deshmukh, J. V. Formal verification and control with conformal prediction: Practical safety guarantees for autonomous systems. *IEEE Control Systems*, 45(6):72–122, 2025.

Liu, Z., Guo, Z., Lin, H., Yao, Y., Zhu, J., Cen, Z., Hu, H., Yu, W., Zhang, T., Tan, J., and Zhao, D. Datasets and benchmarks for offline safe reinforcement learning. *Journal of Data-centric Machine Learning Research*, 2024.

Liu, Z., Li, X., and Zhang, J. C2IQL: Constraint-conditioned implicit q-learning for safe offline reinforcement learning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=97N3XNtFwy.

Mitchell, I. A toolbox of level set methods. *http://www. cs. ubc. ca/mitchell/ToolboxLS/toolboxLS.pdf*, 2004.

Park, S., Li, Q., and Levine, S. Flow q-learning. In *International Conference on Machine Learning (ICML)*, 2025.

Robey, A., Hu, H., Lindemann, L., Zhang, H., Dimarogonas, D. V., Tu, S., and Matni, N. Learning control barrier functions from expert demonstrations. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 3717–3724, 2020. doi: 10.1109/CDC42340.2020.9303785.

So, O. and Fan, C. Solving stabilize-avoid optimal control via epigraph form and deep reinforcement learning. In *Robotics: Science and Systems*, 2023.

Soner, H. M. Optimal control with state-space constraint i. *SIAM Journal on Control and Optimization*, 24(3): 552–561, 1986. doi: 10.1137/0324032. URL https://doi.org/10.1137/0324032.

Stooke, A., Achiam, J., and Abbeel, P. Responsive safety in reinforcement learning by PID lagrangian methods. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9133–9143. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/stooke20a.html.

Sutton, R. S., Barto, A. G., et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Tayal, M., Zhang, H., Jagtap, P., Clark, A., and Kolathaya, S. Learning a formally verified control barrier function in stochastic environment. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pp. 4098–4104, 2024. doi: 10.1109/CDC56724.2024.10886052.

Tayal, M., Singh, A., Kolathaya, S., and Bansal, S. A physics-informed machine learning framework for safe and optimal control of autonomous systems. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=SrfwiloGQF.

Tayal, M., Tayal, M., Singh, A., Kolathaya, S., and Prakash, R. V-OCBF: Learning safety filters from offline data via value-guided offline control barrier functions. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=PGO9mpIyyb.

Wang, H., Dhande, A., and Bansal, S. Cooptimizing safety and performance with a control-constrained formulation. *IEEE Control Systems Letters*, 8:2739–2744, 2024. doi: 10.1109/LCSYS.2024.3511429.

Wang, Z., Hunt, J. J., and Zhou, M. Diffusion policies as an expressive policy class for offline reinforcement learning. *The Eleventh International Conference on Learning Representations*, 2023.

Xu, H., Zhan, X., and Zhu, X. Constraints penalized q-learning for safe offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8753–8760, 2022.

Zhao, W., He, T., Chen, R., Wei, T., and Liu, C. Safe reinforcement learning: A survey. *arXiv preprint arXiv:2302.03122*, 2023.

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. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=cM95sT3gM3.

A. Theoretical Insights

A.1. Derivation of the Epigraph Reformulation

In this section, we formally derive the feasibility epigraph value function and prove its equivalence to the original state-constrained objective. We show that maximizing the constrained value is equivalent to finding the maximal level-set of an unconstrained feasibility value function defined over an augmented state space.

A.1.1. Problem Formulation

Consider a deterministic Markov Decision Process (MDP) with state dynamics $x_{k+1} = f(x_k, a_k)$ and a reward function $r(x_k)$. We enforce a state constraint $\ell(x_k) \geq 0$ for all time steps $k \geq t$.

Let Π denote the set of all admissible policies. The state-constrained value function $V(x_t)$ is defined as the supremum of the cumulative discounted return subject to safety constraints:

$$V(x_t) := \sup_{\pi \in \Pi} \left\{ \sum_{k=t}^{\infty} \gamma^{k-t} r(x_k) \;\middle|\; \ell(x_k) \geq 0, \forall k \geq t \right\}. \tag{18}$$

A.1.2. Epigraph Value Function

We introduce an auxiliary scalar decision variable $z_t \in \mathbb{R}$ (or epigraph variable), which represents a candidate performance threshold (Boyd & Vandenberghe, 2004). We define the *feasibility epigraph value function* $\hat{V}(x_t, z_t)$ as the optimal value of a maximin objective that jointly considers the return margin and the safety margin:

$$\hat{V}(x_t, z_t) := \sup_{\pi \in \Pi} \min \left\{ \sum_{k=t}^{\infty} \gamma^{k-t} r(x_k) - z_t, \quad \inf_{s \geq t} \gamma^{s-t} \ell(x_s) \right\}. \tag{19}$$

Proposition A.1 (Equivalence of Epigraph Formulation)**.** *The original constrained value function $V(x_t)$ can be recovered from the zero-level set of the epigraph feasibility value function $\hat{V}(x_t, z_t)$. Specifically:*

$$V(x_t) = \sup\{z_t \in \mathbb{R} \mid \hat{V}(x_t, z_t) \geq 0\}. \tag{20}$$

Proof. We proceed by transforming the optimization problem in (18) using the epigraph form. Let $J(\pi; x_t) = \sum_{k=t}^{\infty} \gamma^{k-t} r(x_k)$ denote the return and $S(\pi; x_t) = \inf_{s \geq t} \ell(x_s)$ denote the safety margin of a policy. The constraint $\ell(x_k) \geq 0 \; \forall k$ is equivalent to $S(\pi; x_t) \geq 0$.

The original problem can be rewritten by introducing the variable z_t to represent the objective value:

$$\begin{aligned} V(x_t) = \sup_{\pi, z_t} \quad & z_t \\ \text{s.t.} \quad & J(\pi; x_t) \geq z_t, \\ & S(\pi; x_t) \geq 0. \end{aligned} \tag{21}$$

The two inequality constraints must hold simultaneously. This is logically equivalent to requiring the minimum of the constraint residuals to be non-negative. Note that $J(\pi; x_t) \geq z_t \iff J(\pi; x_t) - z_t \geq 0$. Thus, the constraints are satisfied if and only if:

$$\min\{J(\pi; x_t) - z_t,\; S(\pi; x_t)\} \geq 0. \tag{22}$$

Substituting this back into the optimization problem:

$$\begin{aligned} V(x_t) &= \sup_{z_t} \{z_t \mid \exists \pi \in \Pi \text{ s.t. } \min\{J(\pi; x_t) - z_t, S(\pi; x_t)\} \geq 0\} \\ &= \sup_{z_t} \left\{ z_t \;\middle|\; \sup_{\pi \in \Pi} \min\{J(\pi; x_t) - z_t, S(\pi; x_t)\} \geq 0 \right\}. \end{aligned} \tag{23}$$

To align this with the definition in (19), we observe that the sign of the safety term is invariant under multiplication by a positive discount factor $\gamma^{s-t} > 0$. Therefore, $S(\pi; x_t) \geq 0$ is equivalent to $\inf_{s \geq t} \gamma^{s-t} \ell(x_s) \geq 0$.

Replacing the safety term, the inner optimization becomes exactly $\hat{V}(x_t, z_t)$. Thus:

$$V(x_t) = \sup\{z_t \in \mathbb{R} \mid \hat{V}(x_t, z_t) \geq 0\}. \tag{24}$$

□

Remark on Discounted Safety. In (19), the safety constraint $\ell(x_s)$ is scaled by the discount factor γ^{s-t}. Since $\gamma \in (0, 1)$, this scaling does not alter the sign of the term, nor does it change the feasibility boundary (i.e., $\gamma^{s-t}\ell(x_s) \geq 0 \iff \ell(x_s) \geq 0$). This modification is introduced solely to ensure the resulting operator is a contraction, enabling the use of standard discrete-time Bellman recursions for learning $\hat{V}$.

A.2. Theorem (3.1)

Theorem 3.1 (Bounded Epigraph Value Recursion) Consider a deterministic SC-MDP with bounded reward and discount factor $\gamma \in (0, 1)$. Given a transition $(x_t, a_t, r_t, \ell_t, x_{t+1})$ and the epigraph update $z_{t+1} = (z_t - r(x_t))/\gamma$, define $V_r(x_t) = \sup_\pi \sum_{k=t}^{\infty} \gamma^{k-t} r(x_k)$ and $V_s(x_t) = \sup_\pi \min_{s \geq t} \gamma^{s-t}\ell(x_s)$. Then the epigraph value function $\hat{V}(x_t, z_t)$ satisfies:
(i) Bellman recursion.

$$\hat{V}(x_t, z_t) = \min\left\{\ell(x_t),\ \gamma\hat{V}(x_{t+1}, z_{t+1})\right\}.$$

(ii) Decomposition upper bound.

$$\hat{V}(x_t, z_t) \ \leq\ \min\left\{V_r(x_t) - z_t,\ V_s(x_t)\right\}.$$

Proof. We prove each part separately.

Part (i): Bellman recursion. We begin from the definition of the epigraph feasibility value function in (3). For any policy π and initial state x_t, decompose the two terms inside the minimum.

First, the discounted return term admits the decomposition

$$\sum_{k=t}^{\infty} \gamma^{k-t} r(x_k) - z_t = \gamma\Big(\sum_{k=t+1}^{\infty} \gamma^{k-(t+1)} r(x_k) - z_{t+1}\Big), \tag{25}$$

where the relation follows from the definition $z_{t+1} = (z_t - r(x_t))/\gamma$.

Second, the discounted safety term satisfies

$$\min_{s \geq t} \gamma^{s-t}\ell(x_s) = \min\left\{\ell(x_t),\ \gamma \min_{s \geq t+1} \gamma^{s-(t+1)}\ell(x_s)\right\}. \tag{26}$$

Substituting these decompositions into (3) yields

$$\hat{V}(x_t, z_t) = \sup_\pi \min\Big\{\gamma\big(\sum_{k=t+1}^{\infty} \gamma^{k-(t+1)} r(x_k) - z_{t+1}\big),\ \ell(x_t),\ \gamma \min_{s \geq t+1} \gamma^{s-(t+1)}\ell(x_s)\Big\}. \tag{27}$$

Since $\ell(x_t)$ is independent of future actions, and $\gamma > 0$, the minimum can be reordered as

$$\hat{V}(x_t, z_t) = \min\Big\{\ell(x_t),\ \gamma \sup_\pi \min\Big(\sum_{k=t+1}^{\infty} \gamma^{k-(t+1)} r(x_k) - z_{t+1},\ \min_{s \geq t+1} \gamma^{s-(t+1)}\ell(x_s)\Big)\Big\}. \tag{28}$$

Recognizing the inner supremum as the definition of $\hat{V}(x_{t+1}, z_{t+1})$ completes the proof of part (i) and yields the recursion (5): $\hat{V}(x_t, z_t) = \min\left\{\ell(x_t),\ \gamma\hat{V}(x_{t+1}, z_{t+1})\right\}$.

Remark. An analogous Bellman recursion for the epigraph value function in the online (on-policy) setting was independently established by So & Fan (2023); part (i) extends their result to the offline, dataset-driven regime.

Part (ii): Decomposition upper bound. Starting from the definition (3), we apply the standard sup-min inequality ($\sup \min\{A, B\} \leq \min\{\sup A, \sup B\}$) to separate the supremum over policies from the minimum over the two terms:

$$\begin{aligned}\hat{V}(x_t, z_t) &= \sup_{\pi} \min\Big\{ \sum_{k=t}^{\infty} \gamma^{k-t} r(x_k) - z_t,\ \min_{s \geq t} \gamma^{s-t} \ell(x_s) \Big\} \\ &\leq \min\Big\{ \sup_{\pi} \sum_{k=t}^{\infty} \gamma^{k-t} r(x_k) - z_t,\ \sup_{\pi} \min_{s \geq t} \gamma^{s-t} \ell(x_s) \Big\} = \min\{V_r(x_t) - z_t,\ V_s(x_t)\}.\end{aligned} \tag{29}$$

The inequality is strict whenever the policies that individually maximize the return and the safety margin differ, which is generically the case. □

A.3. Bounded Fixed-Point Characterization: Detailed Discussion

This section expands on the paragraph *"Why both parts are necessary"* in Section 3.

Role of part (i): Bellman recursion. The recursion in Theorem 3.1(i) characterizes the fixed-point structure of $\hat{V}$ and drives the value estimate upward toward the tightest feasible boundary supported by the data dynamics. Without this recursion, any function satisfying the upper bound in part (ii) would be consistent — including the trivial solution $\hat{V}(x, z) \to -\infty$, which satisfies the inequality vacuously but is uninformative.

Role of part (ii): decomposition upper bound. Although the recursion characterizes the correct fixed point, it is ill-conditioned for offline learning. Observe that the Bellman target in part (i) is structurally identical to the safety value recursion (equation (7) in the main text): z enters only through z_{t+1}, and the $\min\{\ell, \gamma(\cdot)\}$ operator is the same as for V_s. In practice, neural networks exploit this structural degeneracy and converge to solutions where $\hat{V}(x, z) \approx V_s(x)$ for all z, as demonstrated by the contour plots in Appendix A.4.

When this collapse occurs in safe regions where $V_s(x)$ is large, the correct value should be $\hat{V}(x, z) \approx V_r(x) - z$, which is strictly smaller than $V_s(x)$. A collapsed estimate ignoring z therefore violates the decomposition bound. By enforcing part (ii) as a regularizer, we inject an explicit linear dependence on z through $V_r(x) - z$, preventing the network from disregarding the epigraph variable.

Complementary roles. In summary, part (i) identifies the correct fixed point from *below* (it pushes $\hat{V}$ upward), while part (ii) constrains it from *above* (it prevents collapse). Together they define a bounded fixed-point characterization that is well-conditioned for offline learning with function approximation.

A.4. Effect of the Decomposition Regularizer on Epigraph Sensitivity

This appendix analyzes the effect of the decomposition-based regularizer (equation (11)) on the learned feasibility epigraph value function $\hat{V}(x, z)$. Figure 7 shows contour plots of $\hat{V}$ evaluated at multiple values of z, comparing the unregularized objective with the regularized variant using $\lambda = 0.25$. In the absence of regularization, the learned $\hat{V}_\theta$ exhibits nearly identical contour geometry across different z, indicating weak sensitivity to the epigraph variable. This behavior is consistent with the analysis in Section 3, i.e., although z_t appears in the recursion, the Bellman update does not explicitly enforce conditioning on z, allowing the network to collapse to a representation that largely ignores the epigraph dimension.

Introducing the decomposition regularizer restores meaningful dependence on z. The regularizer enforces the inequality $\hat{V}(x, z) \leq \min\{V_r(x) - z,\ V_s(x)\}$, which injects an explicit linear dependence on z through the $V_r(x) - z$ term. As a result, the learned $\hat{V}_\theta$ must adapt as z varies in order to remain feasible, leading to visibly distinct level sets across different z values in the regularized case. This effect prevents uncontrolled drift along the epigraph dimension and ensures that $\hat{V}_\theta$ encodes a nontrivial performance envelope.

Crucially, restoring sensitivity to z is necessary for reliable extraction of the state-wise threshold $z^*(x) = \sup\{z \mid \hat{V}(x, z) \geq 0\}$. When $\hat{V}_\theta$ is nearly invariant to z, the reward threshold z becomes ill-defined and unstable, undermining policy synthesis. The regularizer therefore plays a structural role as ensures that the learned feasibility value function preserves the intended semantics of the epigraph formulation. Additionally, an ablation study on different values of λ is provided in Section C.3.

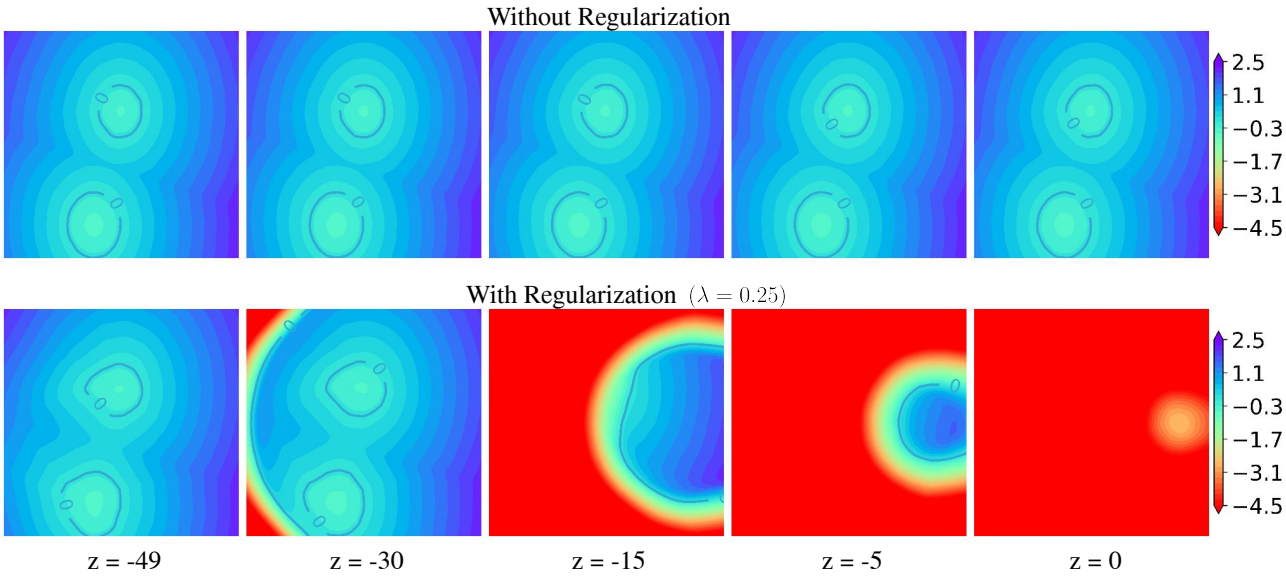

Figure 7. **Feasibility epigraph value function** $\hat{V}_\theta(x, z)$. (*Top*) Learned without the decomposition regularizer, where $\hat{V}_\theta$ exhibits weak sensitivity to the epigraph variable z and similar contour structure across different z values. (*Bottom*) Learned with the regularizer ($\lambda = 0.25$), which restores meaningful dependence on z and yields distinct level sets consistent with the epigraph formulation.

A.5. Reward and Safety Value Function Losses

The decomposition regularizer in equation (11) requires estimates of the optimal reward value $V_r(x)$ and the optimal safety value $V_s(x)$. We learn these using IQL-style (Kostrikov et al., 2022) Q-function and expectile-distilled value losses. The reward value functions Q_r and V_r are trained via:

$$\mathcal{L}_{Q_r}(\phi') = \mathbb{E}\big[(r(x) + \gamma V_r^{\phi}(x') - Q_r^{\phi'}(x, a))^2\big], \tag{30}$$

$$\mathcal{L}_{V_r}(\phi) = \mathbb{E}\big[\mathcal{L}^\tau(Q_r^{\phi'}(x, a) - V_r^{\phi}(x))\big]. \tag{31}$$

Similarly, the safety value functions Q_s and V_s are learned via:

$$\mathcal{L}_{Q_s}(\psi') = \mathbb{E}\big[(\min\{\ell(x), \gamma V_s^{\psi}(x')\} - Q_s^{\psi'}(x, a))^2\big], \tag{32}$$

$$\mathcal{L}_{V_s}(\psi) = \mathbb{E}\big[\mathcal{L}^\tau(Q_s^{\psi'}(x, a) - V_s^{\psi}(x))\big]. \tag{33}$$

All expectations are empirical averages over minibatches from $\mathcal{D}$, and τ is the same expectile parameter used for the feasibility value distillation.

A.6. Expectile Regression

Expectile regression is a classical tool in statistics and econometrics for estimating asymmetric statistics of a random variable. For a random variable X, the τ-expectile is defined as the minimizer of the asymmetric least-squares problem

$$m_\tau = \arg\min_m \mathbb{E}_{x\sim X}\left[L^\tau(x - m)\right], \tag{34}$$

where $L_\tau(y) = |\tau - \mathbf{1}(y < 0)|y^2$.

When $\tau > 0.5$, this loss assigns more weight to samples above the estimate m_τ and less weight to samples below it. Conversely, $\tau < 0.5$ emphasizes lower values. Thus, expectiles interpolate smoothly between the mean ($\tau = 0.5$) and a "high-value–seeking" statistic as $\tau \to 1$.

Expectile regression can also be extended to learning conditional expectiles:

$$m_\tau(x) = \arg\min_{m(\cdot)} \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[L^\tau(y - m(x))\right], \tag{35}$$

which can be optimized efficiently via stochastic gradient descent. This makes expectiles easy to implement in modern machine-learning pipelines, unlike alternative high-order statistics that require specialized solvers.

A.7. Algorithm for Learning the Epigraph Feasibility Value Function

This section covers algorithms corresponding to the procedures explained in Section 3. The algorithm for learning the Epigraph Feasibility Value Function $\hat{V}_\theta(x, z)$ is detailed in Algorithm 1.

Algorithm 1 Learning the Epigraph Feasibility Value Function

Require: Offline dataset $\mathcal{D} = \{(x, a, r, \ell, x')\}$, discount factor γ, expectile parameter τ, regularization weight λ
Initialize networks $\hat{Q}_{\theta'}$, $\hat{V}_\theta$, $Q_r^{\phi'}$, V_r^{ϕ}, $Q_s^{\psi'}$, V_s^{ψ}
while not converged **do**
  Sample minibatch $(x, a, r, \ell, x') \sim \mathcal{D}$
  Sample $z \sim [z_{\min}, z_{\max}]$ and compute $z' = \frac{z - r(x)}{\gamma}$
  Feasibility Q-update:
  $y_Q \leftarrow \min\{\ell(x), \gamma \hat{V}_\theta(x', z')\}$
  Update $\hat{Q}_{\theta'}$ by minimizing $\mathcal{L}_{\hat{Q}}$
  Reward value updates:
  Update $Q_r^{\phi'}, V_r^{\phi}$ using $\mathcal{L}_{Q_r}, \mathcal{L}_{V_r}$
  Safety value updates:
  Update $Q_s^{\psi'}, V_s^{\psi}$ using $\mathcal{L}_{Q_s}, \mathcal{L}_{V_s}$
  Expectile value distillation & regularization:
  Update $\hat{V}_\theta$ using $\mathcal{L}_{\hat{V}} + \lambda \cdot \mathcal{L}_{reg}$
end while
Return Learned epigraph feasibility value function $\hat{V}_\theta$

In succession to the Algorithm 1, next we discuss about a practical algorithm to use this learned $\hat{V}_\theta(x, z)$ for learning a deployable policy (refer Algorithm 2).

Algorithm 2 Flow Matching based Policy Synthesis

Require: Learned $\hat{V}_\theta$, learned $\hat{Q}_{\theta'}$, offline dataset $\mathcal{D}$
Initialize flow teacher μ_ζ and actor μ_ω
for each training iteration **do**
  Sample $(x, a) \sim \mathcal{D}$ and $\eta \sim \mathcal{N}(0, I)$
  Compute $z^\star(x) \leftarrow \textsc{BinarySearch}(\hat{V}_\theta(x, z) \geq 0)$
  Update μ_ζ by minimizing $\mathcal{L}_{\text{flow}}$
  Update μ_ω by minimizing $\mathcal{L}_{\text{actor}}$
end for
Return $\pi^\star(a \mid x) \leftarrow \mu_\omega(x, \eta)$

B. Description of the Experiments

B.1. Boat Navigation

The state $x \in \mathcal{X} = [-3, 2] \times [-2, 2]$ of the 2D Boat system are $x = [x_1, x_2]^T$, where, x_1, x_2 are the x and y coordinates of the boat respectively. We define the step reward at each step, $r(x)$ as:

$$r(x) = C \cdot \left(- \left\| \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} x_{g1} \\ x_{g2} \end{bmatrix} \right\| \right) \tag{36}$$

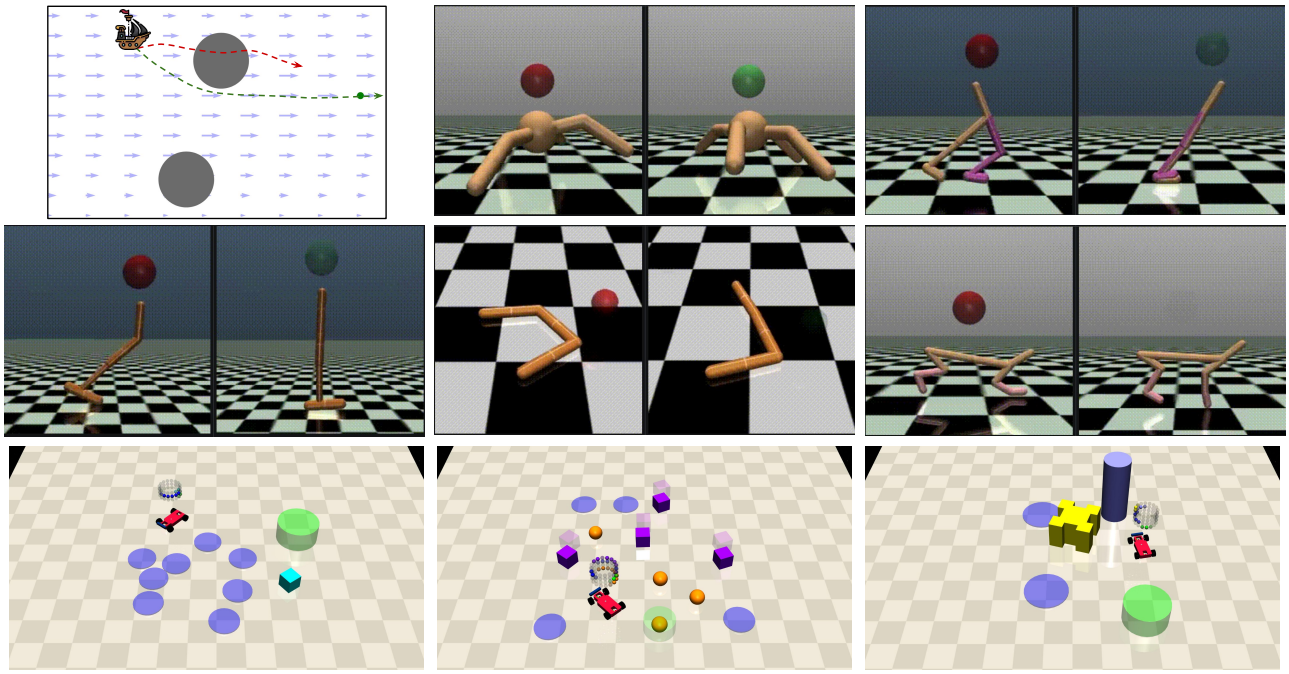

Figure 8. **Illustration of all evaluation environments.** (*Top-Left*) environment illustrates Boat Navigation with obstacles and a goal point. (*Middle*) illustrates SafeVelocity MuJoCo environments with **Red** sphere denoting unsafe state. (*Bottom*) depicts the SafeCarNavigation MuJoCo environments with various obstacles and objectives.

where $[x_{g1}, x_{g2}]^T = [1.5, 0.0]^T$ denotes the goal location, and $C = 0.1$. Maximising this reward drives the boat toward the goal. Consequently, the (augmented) dynamics of the 2D Boat system are:

$$\begin{aligned}(x_1)_{t+1} &= (x_1)_t + (a_1 + 2 - 0.5x_2^2) \cdot \Delta t \\ (x_2)_{t+1} &= (x_2)_t + a_2 \cdot \Delta t \\ z_{t+1} &= (z_t - r(x))/\gamma\end{aligned}$$

where Δt represents discrete time step, a_1, a_2 represent the velocity control action in x_1 and x_2 directions respectively, with $a_1^2 + a_2^2 \leq 1$ and $2 - 0.5x_2^2$ specifies the current drift along the x_1-axis. The safety constraints are encoded as:

$$\ell(x) := min(\|x - (-0.5, 0.5)^T\| - 0.4, \|x - (-1.0, -1.2)^T\| - 0.4) \tag{37}$$

where $\ell(x) < 0$ indicates that the boat is inside a obstacle, thereby ensuring that the sub-level set of $\ell(x)$ defines the failure region.

Offline Data Generation. Since this is a custom environment, we construct an offline dataset which can be used to train a safe and performant policy. We sample 2500 initial states uniformly from $\mathcal{X}$ and simulate each trajectory for 400 discrete timesteps with step size $\Delta t = 0.005s$. During data collection, control inputs are drawn uniformly at random from the admissible range, ensuring diverse system trajectories for learning the controller.

B.2. Safety Gymnasium Environments:

To evaluate scalability to higher-dimensional systems we use Safety Gymnasium (Ji et al., 2023). We consider (i) *safe velocity* variants that impose velocity constraints on Gymnasium MuJoCo-v4 agents (HalfCheetah, Hopper, Ant, Walker2D, Swimmer), and (ii) car navigation tasks (CarButton1, CarGoal1, CarPush1) focused on collision avoidance.

The agent receives a scalar cost signal on the basis of its current state at each step. This cost can be framed as a sparse and simple safety function

$$\ell(x) = \begin{cases} 25 & \text{if cost} = 0, \\ -1 & \text{if cost} > 0. \end{cases} \tag{38}$$

since $\ell(x) \geq 0$ corresponds to safe state. This safety value function provides us a sparse signal.

C. Ablation Studies

We conduct a series of ablation studies to isolate and validate the key design choices of our method. The primary testbed for these analyses is the *Safe Boat Navigation* environment, which provides a controlled yet challenging safety-critical setting. Its low-dimensional, 2D structure enables clear visualization and interpretation of policy behavior, while the presence of nonlinear river drift dynamics makes safe navigation nontrivial for existing baselines. While most ablations are centered on the Boat environment, we additionally include select higher-dimensional systems in specific studies to assess robustness and generalization beyond the primary domain. The following subsections detail these analyses.

C.1. Sensitivity to Expectile parameter (τ)

As described in Section 3, we employ a τ-expectile regression objective to learn the epigraph feasibility value function $\hat{V}(x, z)$, which is subsequently used for policy extraction in the SC-MDP setting. When $\tau = 0.5$, the expectile loss reduces to a symmetric mean-squared error objective, yielding an estimate that reflects the average safety and performance induced by the behavior policy. This *behavior-induced* value function is typically overly conservative and fails to emphasize high-quality safe transitions, which is misaligned with the objectives of safety-constrained control.

To study the effect of asymmetric weighting, we compare $\tau = 0.5$ against larger expectile values that place increasing emphasis on the upper tail of the value distribution. Table 1 reports mean episode reward, safety rate, and average episode cost on the Boat Navigation environment for different τ settings.

Table 1. Mean Reward, Safety Rate (%), and Mean Cost reported for Boat Navigation by varying τ in the expectile regression. ($\pm$) represents standard deviation.

Expectile Parameter (τ)	**0.5 (Behaviour Policy)**	**0.7**	**0.8**	**0.9**
Mean Episode Reward	-485.86 $\pm$ 0.74	-492.36 $\pm$ 0.21	-496.01 $\pm$ 0.58	**-496.10 $\pm$ 0.37**
Safety Rate (%)	88.9 $\pm$ 0.23	98.9 $\pm$ 1.01	99.9 $\pm$ 0.18	**100.0 $\pm$ 0.0**
Mean Episode Cost	1.95 $\pm$ 0.19	0.09 $\pm$ 0.03	0.04 $\pm$ 0.05	**0.0 $\pm$ 0.0**

We observe that setting $\tau = 0.5$ leads to the lowest safety rate, as the resulting value function underweights safety-critical regions that are infrequently represented in the offline dataset. As τ increases, safety performance improves steadily while reward remains competitive, reflecting stronger alignment with high-value demonstrated actions. Consequently, larger expectile values produce policies that more effectively capture the upper envelope of safe and performant behaviors.

In idealized settings, such as deterministic dynamics with minimal dataset noise, pushing $\tau \to 1$ concentrates the estimator on the extreme upper tail and can lead to the strongest empirical performance by prioritizing high-value next-state targets. However, this regime is brittle in practice as rare suboptimal actions or stochastic disturbances may generate overly optimistic transitions, causing extreme expectile estimators to overfit spurious outcomes.

Balancing this trade-off, we fix $\tau = 0.9$ in all experiments. This choice substantially up-weights high-value targets while preserving robustness to noise and outliers commonly present in offline datasets.

C.2. Resilience to External Perturbations

We next evaluate the robustness of the recovered policy to perturbations in the executed control action, which indirectly probes sensitivity to errors in the learned feasibility value function. Specifically, we perturb the selected optimal action a by additive zero-mean Gaussian noise $\mathcal{N}(0, \sigma^2)$, with σ scaled relative to the maximum admissible action magnitude. We consider three perturbation levels: 5%, 10%, and 20% of the action bound, and evaluate performance over 100 independent rollouts per setting, using five random seeds. In addition to the Boat Navigation task, we also include two higher-dimensional systems from Safety Gymnasium, namely Safe Velocity HalfCheetah and Ant, to assess robustness beyond the primary environment.

Figure 9 summarizes reward and safety cost trends across perturbation levels. Across all environments, safety costs increase only mildly under small to moderate perturbations, indicating that the learned policies tolerate reasonable inaccuracies in

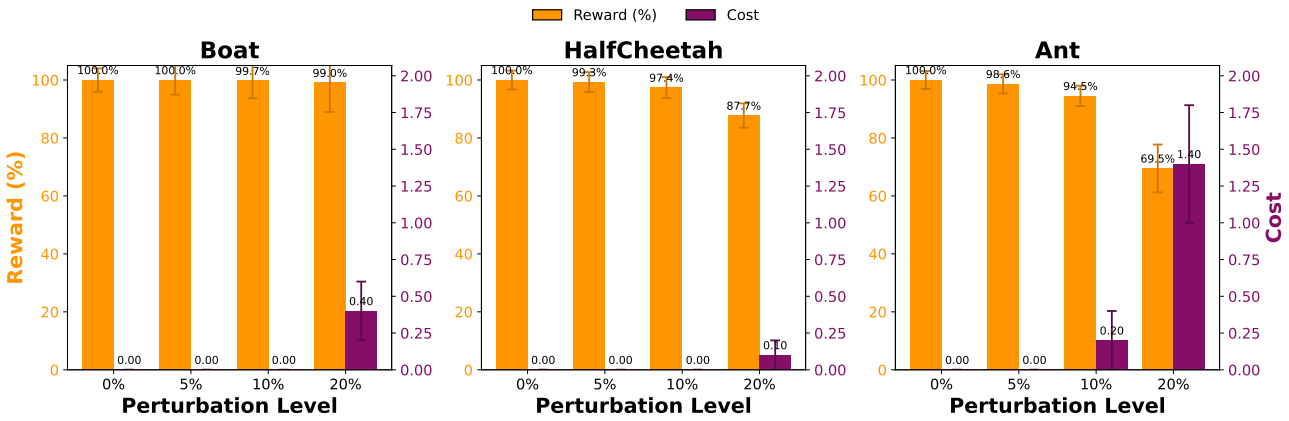

Figure 9. **Results for Control under Perturbations.** The plots show average reward (%) ($\pm$ standard deviation) achieved relative to no perturbations case, at different levels of perturbations (5%, 10% and 20%), and their average absolute cost values ($\pm$ standard deviation) respectively.

control execution even in safety-critical regimes. While rewards degrade more noticeably at larger perturbation magnitudes, this degradation becomes significant only under aggressive noise injection. These results suggest that precise action execution improves performance, but our framework retains strong safety-aware behavior and competitive returns under moderate control uncertainty. Future work will investigate uncertainty-aware and robust optimization formulations that explicitly model and mitigate the effects of disturbances.

C.3. Effect of the Regularizer Parameter λ on $\hat{V}_\theta$

We continue from the ablation study on regularizer parameter in section 4.3 of the main paper. Figure 10 visualizes the inferred per-state budget $z^*(x)$ (equation (13)) for the Boat Navigation task. Based on the figure there seems a bias–variance like trade-off across the sweep. For small regularizer parameter value ($\lambda = 0.1$) the decomposition constraint is too weak as $\hat{V}_\theta$ still exhibits low sensitivity to z, causing inflated estimates of high reward feasibility (red region towards left-end in $\lambda = 0.1$ plot), which means that even for realistically low reward yielding regions, $\hat{V}_\theta$ suggests higher reward budget z than is possible. However, as λ increases, the decomposition term more strongly enforces the upper bound $\hat{V} \leq \min\{V_r - z,\, V_s\}$, which brings the z-budget over-estimates within reasonable ranges. But increasing the λ too much can also drift $\hat{V}_\theta$ away from the original recursion objective.

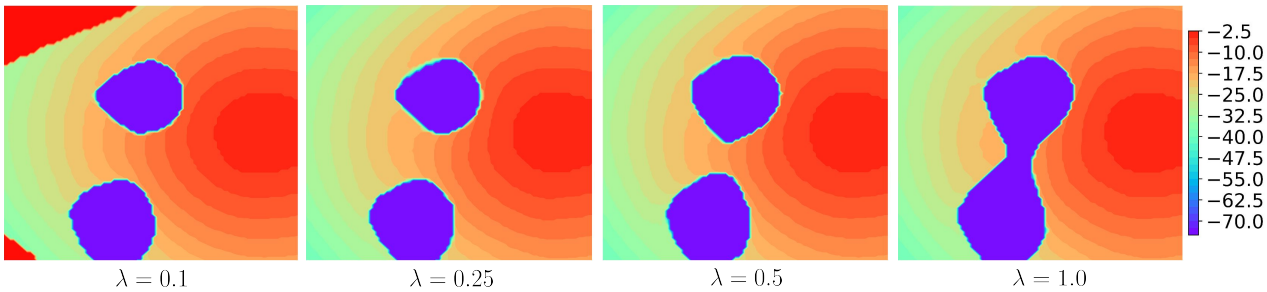

Figure 10. **Effect of the regularizer parameter λ on the inferred epigraph budget.** For the Boat Navigation environment, each panel shows the state-wise optimal budget $z^*(x)$ defined by the feasibility condition $\hat{V}_\theta(x, z^*) \geq 0$. As λ increases, the feasible region contracts and $z^*(x)$ becomes more conservative; states shaded in purple correspond to infeasible regions.

Based on the results from the plot in Figure 4 and balancing the above discussed effects, we select $\lambda = 0.25$ for the main experiments as it sufficiently prevents the z-desensitization and budget-collapse as in case no regularization, while also avoiding excessive drifting away from original objective as seen at higher weights. Practitioners should tune λ on a validation set (a useful range is roughly 0.1–0.5) and inspect both sensitivity diagnostics (e.g., variation of $\hat{V}_\theta$ across z (Appendix A.4)) and downstream trade-offs between reward and safety when selecting the final value.

C.4. Flow Matching v/s Diffusion Model

In Section 3.3, where we discussed about synthesizing our policy using Flow Q-Learning (Park et al., 2025), a question arises, whether the performance gains in our proposed framework just due to the use of Flow Matching based policy learning or is the use of Epigraph Feasibility Value Function actually adding any value. To study this, we perform the analysis where we compare the results of our proposed framework with 2 more policy learning Approaches, Diffusion Q-Learning (Wang et al., 2023) and Weighted Behavior Cloning.

Table 2. Mean Episode Reward (R) and Mean Episode Cost (C) ($\pm$ standard deviation) reported for proposed framework across different policy backbones.

Metric	Flow Matching	Diffusion Model	Weighted BC
R	-496.23 $\pm$ 0.34	-584.53 $\pm$ 2.37	-716.60 $\pm$ 5.23
C	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0	18.56 $\pm$ 4.45

Table 2 shows results for these policy learning approaches when plugged into our framework. It is evident that although weighted Behavior Cloning fails miserably to produce good results due to lack of its expressiveness (Wang et al., 2023; Park et al., 2025), but when Diffusion Q-Learning is implemented, the results still show an improvement compared to the other baselines mentioned in the main paper (Figure 3). Overall, Flow Q-Learning based policy outperforms Diffusion Q-Learning in terms of performance, which is also why we prefer to use Flow Q-Learning for our framework.

D. Experimental Details

D.1. Experimental Hardware

To keep the evaluation fair and avoid any discriminatory added advantage to any specific experiment, all experiments were conducted on a single system equipped with an 14th Gen Intel Core i9-14900KS CPU, 128GB RAM, and an NVIDIA GeForce RTX 4090 GPU for training and experiment evaluations.

D.2. Network Architecture and Training Details of the Proposed Algorithm

We have compiled and listed down all the hyperparameters that we used to perform our experiments and report the results. These training settings for all the environments are detailed in the Table 3. For the MuJoCo environments, we use the widely accepted DSRL (Liu et al., 2024) dataset.

During the training, we set the τ for expectile regression in Apppendix C.1 to 0.9. And we use clipped double Q-learning (Fujimoto et al., 2018), taking a minimum of two Q. We update the target Q networks using Exponential Moving Average (EMA) where the weight to the new parameters is set to 0.001. Following (Kostrikov et al., 2022), we clip exponential advantages to $(-\infty, 100]$ in feasible part and $(-\infty, 150]$ in infeasible part. For our paper, we used flow matching implementation from FQL (Park et al., 2025).

D.3. Hyperparameters for the Baselines

Hyperparameters for the baselines are provided in Table 4. We use the official implementations of BEAR-Lag, BCQ-Lag, CPQ and COptiDICE from (Liu et al., 2024), of C2IQL from (Liu et al., 2025) of FISOR from (Zheng et al., 2024) of V-OCBF from (Tayal et al., 2026) and of NCBF from (Robey et al., 2020), and train all methods on the same DSRL datasets.

Table 3. Hyperparameters for the Algorithm.

Hyperparameter	**Value**
Value Function Network (VFN)	Multi-Layer Perceptron (MLP)
Activation Function	ReLU
Optimizer	Adam optimizer
Learning Rate	3×10^{-4}
Discount Factor (γ)	0.99
Time Step Intervals (FM Policy)(μ_ζ)	5 (refer Flow Q-Learning)
Boat Navigation	
Number of Layers in VFN	2
Hidden Layer Size in VFN	256 neurons per layer
Dataset Size	1M
Safe Velocity Hopper	
Number of Hidden Layers in VFN	3
Hidden Layer Size in VFN	256 neurons per layer
Dataset Size	1.32M
Safe Velocity Half-Cheetah	
Number of Hidden Layers in VFN	3
Hidden Layer Size in VFN	128 neurons per layer
Dataset Size	249K
Safe Velocity Ant	
Number of Hidden Layers in VFN	3
Hidden Layer Size in VFN	256 neurons per layer
Dataset Size	2.09M
Safe Velocity Walker2D	
Number of Hidden Layers in VFN	3
Hidden Layer Size in VFN	256 neurons per layer
Dataset Size	2.12M
Safe Velocity Swimmer	
Number of Hidden Layers in VFN	3
Hidden Layer Size in VFN	256 neurons per layer
Dataset Size	1.68M
Safe Car Button	
Number of Hidden Layers in VFN	3
Hidden Layer Size in VFN	256 neurons per layer
Dataset Size	2.65M
Safe Car Goal	
Number of Hidden Layers in VFN	3
Hidden Layer Size in VFN	256 neurons per layer
Dataset Size	1.67M
Safe Car Push	
Number of Hidden Layers in VFN	3
Hidden Layer Size in VFN	256 neurons per layer
Dataset Size	2.87M

Table 4. Detailed Hyperparameters for Baseline Special Networks. (Layers, Units) notation refers to hidden layers and units per layer. All baselines utilize the DSRL dataset standards (Liu et al., 2024).

Baseline	Network Component	Architecture Specification
C2IQL (Liu et al., 2025)	Actor / Critic / Value Nets	MLP, (256, 2)
	Cost Reconstruction Model	**MLP, (5 layers, 512 units each)**
	Reward / Cost Advantage	MLP, (256, 2)
BCQ-Lag (Lag. dual of (Fujimoto et al., 2019))	**VAE (Encoder/Decoder)**	**MLP, (750, 2), Latent Dim: 2 $\times$ Action Dim**
	Perturbation Model ξ	MLP, (400, 2)
	Double Q-Critics	MLP, (400, 2)
CPQ (Xu et al., 2022)	Actor (Policy) Net	MLP, (256, 2)
	Constraint-Penalized Q	**Ensemble DoubleQCritic, (256, 2)**
	Cost Critic Net	MLP, (256, 2)
FISOR (Zheng et al., 2024)	Diffusion Denoiser (Actor)	DiffusionDenoiserMLP, (256, 2)
	Feasibility Classifier	**MLP, (256, 2), Sigmoid Output**
	Energy/Value Guidance	MLP, (256, 2)
COptiDICE (Lee et al., 2022)	**Dual / ν Network**	**DualNet (Value-like), (256, 2)**
	Actor (Extraction) Net	MLP, (256, 2)
BEAR-Lag (Lag. dual of (Kumar et al., 2019))	**VAE (Support Model)**	**MLP, (750, 2)**
	Actor (Policy) Net	SquashedGaussianMLPActor, (256, 2)
	Cost / Reward Critics	MLP, (256, 2)
V-OCBF (Tayal et al., 2026)	Value Guided Neural CBF	MLP, (256, 2)
NCBF (Robey et al., 2020)	Neural CBF	MLP, (256, 2)