Spend Smart, Learn Fast: Budget Curricula for Safe RL

Michael Rayan
m.a.rayan@student.tue.nl
[TU/e MSc DS and AI student]
Eindhoven, Netherlands

Thomás de los Santos Verrijp
t.d.los.santos.verrijp@student.tue.nl
[TU/e MSc DS and AI student]
Eindhoven, Netherlands

Sohail Farkish
s.farkish@student.tue.nl
[TU/e MSc DS and AI student]
Eindhoven, Netherlands

Andrei Popoviciu
a.popoviciu@student.tue.nl
[TU/e MSc DS and AI student]
Eindhoven, Netherlands

Francesco Colasurdo
f.colasurdo@student.tue.nl
[TU/e MSc DS and AI student]
Eindhoven, Netherlands

Abstract

Safety remains a core challenge in reinforcement learning (RL), where agents must optimize performance without violating constraints. Traditional Constrained Markov Decision Processes (CMDPs) enforce static safety thresholds, limiting adaptability as agents learn. Budgeted Markov Decision Processes (BMDPs) address this by embedding dynamic safety budgets into the state. However, prior work, notably [6], rely on random budget sampling that leaves the exploration–safety trade-off uncontrolled. We introduce **budget curricula**, structured schedules that regulate how the initial safety budget evolves during training, and integrate them into the Budgeted Fitted Q-Iteration framework. Experiments across continuous control (*Highway-Env*) and discrete dialogue (*Slot-Filling*) domains reveal that curriculum design meaningfully shapes the safety–performance balance. In *Highway-Env*, linear relaxed-to-strict schedules slightly improve average returns over the baseline while keeping violation rates unchanged, whereas exponential relaxed-to-strict schedules trade stability for faster learning. In *Slot-Filling*, linear relaxed-to-strict curricula achieve lower costs and equal constraint satisfaction, suggesting improved efficiency in discrete settings. These results demonstrate that structured budget scheduling provides a controllable mechanism for steering learning behavior in safe reinforcement learning.

Keywords

Safe reinforcement learning, budgeted Markov decision processes (BMDP), constrained MDPs (CMDP), budgeted fitted Q-iteration (BFTQ), curriculum learning, safety budget scheduling

1 Introduction

Ensuring safety remains one of the central challenges in reinforcement learning (RL)[4, 8, 9], especially in real-world environments where unsafe behavior can have severe consequences. Research in safe RL has broadly followed two main directions [8]:

(1) modifying the agent's optimization criterion to explicitly incorporate safety or risk constraints;
(2) shaping the agent's exploration process to avoid unsafe regions during training.

This paradigm offers stronger theoretical guarantees by ensuring constraint satisfaction throughout deployment. Both approaches seek a balance between performance and safety, yet differ in how safety is enforced. In this work, we focus on the first family of methods: *constrained reinforcement learning* (CRL), which introduces safety directly into the optimization objective [22]. A central formalism in CRL is the *Constrained Markov Decision Process* (CMDP) [2], where safety constraints are embedded into the environment's dynamics rather than applied externally. CMDPs require the agent to jointly optimize reward and constraint satisfaction, providing a principled way to trade off performance and safety. Prominent solution classes include Lagrangian-based methods that dynamically balance reward and cost via penalty multipliers [1, 7], and projection-based methods that project updates into the feasible set to maintain safety [14]. Beyond enforcing constraints, CMDPs have also been used for direct behavior specification, where indicator-based cost functions describe desired outcomes such as "avoid lava at least 99% of the time" [18]. However, these formulations rely on static thresholds that limit adaptability as the agent learns.

We depart from this static view by introducing dynamic constraint thresholds that evolve during training. This problem is modeled with *Budgeted Markov Decision Processes* (BMDPs) [5], where the constraint threshold is treated as a budget integrated into the state space. Unlike CMDPs, BMDPs naturally accommodate dynamic budgets, enabling budget-aware policies that adapt to changing safety levels. Carrara et al. [6] demonstrated the feasibility of this formulation in continuous domains, but their random budget sampling rendered the exploration–safety trade-off implicit and uncontrollable, yielding inconsistent policies and limited designer control. To address this limitation, we introduce *structured budget evolution* through curriculum learning. Instead of sampling budgets uniformly, we schedule how the initial budget evolves over training. Curriculum learning has been successfully used in safe RL to guide exploration [11, 19, 21, 23], yet existing works are constrained to CMDPs, where threshold changes require reformulating the environment. In contrast, BMDPs embed the budget within the state, allowing efficient and flexible curricula. We propose and analyze two complementary budget curricula:

(1) **Conservative-to-liberal:** training begins with strict budgets that gradually increase, emphasizing safe behavior early before expanding toward higher-risk, high-reward regimes.
(2) **Liberal-to-conservative:** training starts with generous budgets and progressively tightens them, fostering fast learning of performant strategies followed by refinement toward safer operation.

By comparing these approaches, we study how structured budget evolution influences the safety–performance trade-off and whether curriculum-based adaptation yields agents that are simultaneously robust and high-performing.

Contributions: This paper introduces a framework for controlled budget initialization in BMDPs, bridging static constraint formulations and adaptive, curriculum-driven safe RL. Specifically, we:

- integrate curriculum learning with budgeted reinforcement learning;
- propose several budget curricula and analyze their effects on the safety–performance trade-off;
- provide empirical evidence that structured budget evolution influences (though not always improves) both robustness and learning efficiency compared to random budget initialization.

2 Related Work

Safe RL in the CMDP setting has been explored through multiple constraint-handling techniques. Constrained Policy Optimization [1] extends trust-region methods to enforce near-constraint satisfaction, while primal–dual approaches [20] use Lagrangian multipliers to balance reward and cost dynamically. Although effective, these methods assume fixed thresholds, making them unsuitable when safety requirements change during training.

To enhance interpretability and stability, [18] proposed direct behavior specification with indicator-based cost functions, providing an intuitive, scalable alternative to reward engineering. Yet their approach retains static thresholds, leaving the handling of evolving safety constraints unexplored. BMDPs address this limitation by embedding the constraint threshold (budget) into the state, enabling explicit adaptation to varying safety levels. Carrara et al. [6] developed the first BMDP algorithms for continuous domains, introducing the Budgeted Bellman operator and propagating both reward and cost signals. Their method, however, samples budgets uniformly, without examining whether structured sequencing can accelerate learning or reduce unsafe exploration.

Beyond online RL, the Constrained Decision Transformer (CDT) [15] reformulates safe RL as a multi-objective optimization from offline datasets, allowing dynamic trade-offs at deployment. Nevertheless, CDT assumes fixed thresholds during training and does not investigate curricula over evolving constraints. Curriculum learning itself has long been used to improve RL performance by structuring task difficulty [3], later extended to safe RL to accelerate convergence [17] and promote safer exploration [19]. Yet existing curriculum-based safe RL methods remain confined to CMDPs, where modifying constraint thresholds entails redefining the problem. This significantly limits scalability. To date, no prior work has combined the flexibility of budget-aware formulations with curriculum-driven training to investigate how structured budget sequences affect both safety and learning dynamics.

3 Background and Motivation

All notations for the upcoming sections can be found in section A.

3.1 Budgeted Reinforcement Learning

In Budgeted Reinforcement Learning, we rely on the framework of Budgeted Markov Decision Processes (BMDPs), which can be viewed as an extension of Constrained Markov Decision Processes (CMDPs) by explicitly incorporating a budget variable into the state and decision process.

A Constrained Markov Decision Process (CMDP) extends a Markov Decision Process (MDP) with the concept of costs and associated constraints [12]. Formally, a CMDP is defined as:

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

where $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, $P(s'|s,a)$ the transition probability, $r(s,a)$ the reward function, $c^{(i)}(s,a)$ the i-th cost function for $i = 1, \dots, m$, and $\gamma \in [0,1)$ the discount factor. Although multiple cost functions can model complex multi-objective constraints, in practice using a single cost function often simplifies both the optimization problem and its interpretation.

For a stationary policy π, the expected discounted reward and cost returns are

$$\begin{aligned} J(\pi) &= \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right], \\ J_c^{(i)}(\pi) &= \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t c^{(i)}(s_t, a_t)\right]. \end{aligned} \tag{1}$$

The CMDP optimization problem is then formulated as:

$$\begin{aligned} \max_{\pi} \quad & J(\pi) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right], \\ \text{s.t.} \quad & J_c^{(i)}(\pi) \le d_i, \quad i = 1, 2, \dots, m. \end{aligned} \tag{2}$$

where each d_i is a specified constraint threshold. The feasible policy set is defined as

$$\Pi_{\text{safe}} = \{\pi \mid J_c^{(i)}(\pi) \le d_i, \ \forall i\}.$$

One could also express all constraints through an auxiliary cost function $C(s,a)$ with an upper bound β on its expected return. This leads to the following optimization problem:

$$\begin{aligned} \max_{\pi} \quad & \mathbb{E}_\pi\left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t)\right], \\ \text{s.t.} \quad & \mathbb{E}_\pi\left[\sum_{t=0}^{T} \gamma^t C(s_t, a_t)\right] \le \beta. \end{aligned} \tag{3}$$

A Budgeted Markov Decision Process (BMDP) extends a CMDP by allowing the budget β to evolve dynamically as part of the system's state. Formally, we follow the notation introduced by Carrara *et al.* [6] and define the BMDP as a multi-objective MDP with augmented (budgeted) state and action spaces:

$$\bar{\mathcal{S}} = \mathcal{S} \times \mathcal{B}, \quad \bar{\mathcal{A}} = \mathcal{A} \times \mathcal{B}. \tag{4}$$

The augmented dynamics are defined by:

$$\bar{P}((s', \beta') \mid (s, \beta), (a, \beta_a)) \overset{\text{def}}{=} P(s' \mid s, a)\, \delta(\beta' - \beta_a), \tag{5}$$

so that the policy $\pi(a, \beta_a \mid s, \beta)$ selects both an environment action and the next-step budget. Within an episode, the budget evolves as $\beta_{t+1} = \beta_a$.

The reward and cost functions are stacked into a vector-valued signal:

$$R(s,a) \overset{\text{def}}{=} \begin{bmatrix} R_r(s,a) \\ R_c(s,a) \end{bmatrix} \in \mathbb{R}^2. \tag{6}$$

For any initial augmented state (s_0, β_0), the reward and cost returns of a budgeted policy π are:

$$G_r^\pi = \sum_{t \geq 0} \gamma^t R_r(s_t, a_t), \qquad G_c^\pi = \sum_{t \geq 0} \gamma^t R_c(s_t, a_t), \tag{7}$$

and the budgeted feasibility constraint requires $\mathbb{E}[G_c^\pi \mid \tilde{s}_0] \leq \beta_0$ The value functions for a given policy π are defined as:

$$V^\pi(s) \overset{\text{def}}{=} \mathbb{E}[G^\pi \mid s_0 = s], \ \ Q^\pi(s, a) \overset{\text{def}}{=} \mathbb{E}[G^\pi \mid s_0 = s, a_0 = a] \tag{8}$$

where $V^\pi(s)$ represents the expected return starting from state s, and $Q^\pi(s, a)$ the expected return when taking action a in s and following policy π thereafter. Both functions have two components, one for reward and one for cost:

$$V^\pi = (V_r^\pi, V_c^\pi), \qquad Q^\pi = (Q_r^\pi, Q_c^\pi).$$

The set of feasible policies under a given budget β is defined as:

$$\Pi_a(s) \overset{\text{def}}{=} \{\pi \in \Pi \mid V_c^\pi(s) \leq \beta\}, \tag{9}$$

where $\Pi_a(s)$ denotes the subset of policies whose expected cumulative cost from state s does not exceed the available budget β.
Within this set, the optimal value and policy functions are obtained through the following nested optimizations:

$$V_r^*(s) \overset{\text{def}}{=} \max_{\pi \in \Pi_a(s)} V_r^\pi(s), \qquad \Pi_r(s) \overset{\text{def}}{=} \arg \max_{\pi \in \Pi_a(s)} V_r^\pi(s), \tag{10}$$

$$V_c^*(s) \overset{\text{def}}{=} \min_{\pi \in \Pi_r(s)} V_c^\pi(s), \qquad \Pi^*(s) \overset{\text{def}}{=} \arg \min_{\pi \in \Pi_r(s)} V_c^\pi(s). \tag{11}$$

Budgeted Bellman operator. We learn $Q = (Q_r, Q_c)$ over $\tilde{\mathcal{S}} \times \tilde{\mathcal{A}}$ via the *Budgeted Bellman Optimality Equation*:

$$Q^*(s, a) = R(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s' \mid s, a) \sum_{a' \in \mathcal{A}} \pi_{\text{greedy}}(a' \mid s'; Q^*)\, Q^*(s', a'). \tag{12}$$

The greedy policy π_{greedy} is obtained as the solution of a nested constrained optimization problem:

$$\pi_{\text{greedy}}(a \mid s; Q) \in \arg \min_{\rho \in \Pi_r^Q} \mathbb{E}_{a \sim \rho}[Q_c(s, a)], \tag{13}$$

$$\Pi_r^Q \overset{\text{def}}{=} \arg \max_{\rho \in \mathcal{M}(\mathcal{A})} \mathbb{E}_{a \sim \rho}[Q_r(s, a)] \quad \text{s.t.}\ \mathbb{E}_{a \sim \rho}[Q_c(s, a)] \leq \beta. \tag{14}$$

Intuitively, the inner maximization seeks the action (or mixture of actions) that maximizes expected reward while respecting the current budget β, and the outer minimization selects among those reward-optimal actions the one with the lowest expected cost. In practice, π_{greedy} corresponds to a policy that lies on the convex hull of the (Q_c, Q_r) frontier, ensuring feasibility with respect to the budget.

For implementation purposes, we equivalently express this relationship through the budgeted optimality operator:

$$\mathcal{T} Q(\tilde{s}, \tilde{a}) \ = \ R(\tilde{s}, \tilde{a}) + \gamma\, \mathbb{E}_{\tilde{s}' \sim \tilde{P}} \big[Q\big(\tilde{s}',\ \pi_{\text{hull}}(\tilde{s}'; Q)\big) \big], \tag{15}$$

where π_{hull} implements the above π_{greedy} by interpolating actions along the convex hull of the (Q_c, Q_r) trade-off curve.

BFTQ (learning rule). Budgeted Fitted Q-Iteration (BFTQ) [6] approximates the fixed point of $\mathcal{T}$ via regression on a dataset $\mathcal{D}$ of transitions:

$$\theta_{k+1} = \arg \min_\theta \sum_{(\tilde{s}, \tilde{a}, r, c, \tilde{s}') \in \mathcal{D}} \left\| Q_\theta(\tilde{s}, \tilde{a}) - \hat{\mathcal{T}}\, Q_{\theta_k}(\tilde{s}, \tilde{a}, \tilde{s}') \right\|_2^2. \tag{16}$$

We use ε-greedy exploration with a schedule over episodes, mixing a random budgeted policy and the current greedy policy.

While a CMDP enforces a global constraint on the expected cumulative cost, a BMDP integrates the budget directly into the decision process, enabling adaptive, budget-aware policies that explicitly trade off reward and cost at each step. A key advantage of this formulation is that it enables a single policy to generalize across different safety levels. Unlike CMDPs, which must be reformulated whenever constraint thresholds change, BMDPs naturally handle variable budgets, supporting both static and adaptive safety requirements. This property is particularly relevant in environments where constraints evolve during training or deployment, such as in robotic control, energy management, or autonomous driving.
A RL method for solving BMDPs was introduced by [6] through the Budgeted Fitted Q-Iteration (BFTQ) algorithm. Fitted Q-Iteration uses sampled transitions to iteratively approximate the Bellman target; since the Bellman operator is a contraction mapping, this procedure converges to the optimal Q-function given sufficient samples. In BMDPs, this operator is extended to act over two value dimensions consisting of reward and cost rather than one. The learning objective becomes minimizing a regression loss over both dimensions, which can be achieved with linear approximators, regression trees, or neural networks.

As seen in [6], to collect $\mathcal{D}$, the dataset of budgeted transitions collected during exploration, a risk-sensitive exploration strategy mixes a random budgeted policy π_{rand} and a greedy policy π_{greedy}. Each episode begins with an initial budget β_0 sampled uniformly:

$$\beta_0 \sim \mathcal{U}(\mathcal{B}), \tag{17}$$

so as to expose the agent to the full spectrum of risk levels rather than biasing exploration toward purely reward-seeking behavior. At each step, only feasible action–budget pairs are considered and the budget component is drawn uniformly from the simplex of admissible choices (those satisfying $\mathbb{E}[\beta_a] \leq \beta$), which implements a risk-sensitive exploration strategy while respecting the instantaneous budget constraint.

3.2 Curriculum Learning for Reinforcement Learning

Our next domain of focus is Curriculum Learning [3], which introduces a principled approach to structuring experience acquisition during agent training. The main idea is to control *how* and *in what order* experience is presented to the agent, thereby improving efficiency and stability of learning.
In Reinforcement Learning (RL), a curriculum can be interpreted as an ordered sequence of tasks or samples designed to shape the evolution of difficulty or complexity across training [16]. In our specific case, each “task” corresponds to a distinct initial budget value β_0 (see Section 4). Rather than sampling budgets uniformly at random, we define a structured budget schedule over time namely linear, exponential, or cyclic to study how such ordering affects

learning.
The concept of a curriculum draws inspiration from human and animal learning: we typically master simpler tasks before progressing to more complex ones. An *increasing* curriculum mirrors this behavior, starting with easy, low-risk conditions and gradually exposing the agent to harder, riskier environments.

3.2.1 Core Components of Curriculum Learning. Following [16], a curriculum is defined by three key components:

- Task Generation : This component defines which tasks or conditions the agent will face. In our case, task generation corresponds to defining the different initial budget values β_0 encountered across episodes.
- Sequencing : Sequencing determines the order in which tasks are presented. In our setting, sequencing defines whether the initial budget increases, decreases, or oscillates over time and at what rate.
- Transfer : This component captures how knowledge from one task transfers to the next.

Together, these components structure a progression that guides the agent from safer, simpler experiences to more complex, risk-sensitive ones, promoting stability and generalization.

3.2.2 Types of Curricula: Static and Dynamic. Narvekar et al. distinguish between two main categories of curricula:

- **Static curriculum:** the sequence of tasks (or budgets) is predefined and does not depend on the agent's performance.
- **Dynamic curriculum:** the difficulty evolves based on the agent's success, allowing adaptive progression.

In our BMDP context, this distinction translates to how the initial budget β_0 is determined:

- In a *static* curriculum, β_0 follows a fixed, predefined schedule.
- In a *dynamic* curriculum, β_0 depends on performance metrics from previous episodes.

It is important to note that this adaptability affects only the initialization of each episode; within an episode, the budget evolves step-by-step according to BMDP dynamics.

4 Problem statement: Curriculum Learning for BMDPs

Our main contribution lies in integrating Curriculum Learning principles within the BMDP framework.
While uniform sampling of β_0 promotes broad coverage (eq. 17), it introduces inefficiencies: episodes may begin at infeasible or uninformative budget values, wasting training samples and potentially destabilizing learning.
Alternatively, we modify the *BFTQ* algorithm such that the initial budget β_0 at the beginning of each episode is not sampled uniformly, but rather determined by a structured or adaptive *curriculum schedule.*

In this setting, the budget itself becomes both the constraint and the single "task" operated on by the curriculum. We explore multiple curriculum configurations with each defining a different ordering or evolution of β_0 to address our central research question:

> *Can the ordering of initial budgets influence the agent's learning process? If so, how and to what extent?*

This section establishes the motivation for using curriculum strategies in the BMDP setting, whereas the next section details the ways in which they may be implemented.

5 Methods

To address the question raised in section 4, we propose to replace the random initialization of β_0 with a structured *curriculum function*

$$\mathrm{C} : \mathbb{N} \to \mathrm{B}, \qquad \beta_0^{(e)} = \mathrm{C}(e), \tag{18}$$

which deterministically schedules the episode's starting budget $\beta_0^{(e)}$ based on its index e. This provides a controlled and interpretable progression of budget values whether it is gradually increasing, decreasing, or cyclic, allowing the agent to experience different safety levels in a structured manner. This function can be seen in use by Algorithm 1. In the case of the unmodified algorithm from [6], the same function would have taken the form:

$$\mathrm{C}(e) = \mathrm{Uniform}(\beta_{\min}, \beta_{\max})$$

This section presents our methodological framework for integrating curriculum learning into Budgeted Reinforcement Learning (BRL). We study several curriculum types, each specifying a distinct evolution of β_0 over episodes:

- **Increasing curriculum:** the budget starts small (restrictive) and gradually increases, allowing the agent to master safe behavior before exploring riskier strategies.
- **Decreasing curriculum:** the budget starts large and progressively shrinks, forcing the agent to become more efficient as safety margins tighten.

For example, a *decreasing* curriculum could represent a startup with an initial investment that diminishes over time, requiring increasingly efficient decisions. Conversely, an *increasing* curriculum could represent a system that begins under strict safety constraints and earns more flexibility as it performs well, mirroring the gradual trust humans gain in autonomous systems.

Beyond addressing our research question, these curricula aim to enable smooth adaptation to dynamic constraints, and establish a structured connection between CMDPs, BMDPs, and Curriculum Learning. In essence, the curriculum provides a principled framework for organizing the agent's exposure to varying risk regimes during training, offering new insights into how the sequencing of safety budgets shapes learning and performance.

During training, exploration actions (a, β_a) are sampled uniformly from a feasible simplex $\Delta_{\mathcal{A}\times\mathcal{B}}$ satisfying $\mathbb{E}[\beta_a] \leq \beta$, while exploitation relies on π_{greedy}.

5.1 Curriculum on the Initial Budget

By systematically varying the budget, the agent can progressively adapt from conservative to risk-tolerant behaviors (or vice versa), promoting both safety and efficiency in exploration.

We explore three curriculum schedules, each projected to $[\beta_{\min}, \beta_{\max}]$:

Linear (increasing / decreasing).

$$\begin{aligned} \mathrm{C}(e) &= \beta_{\min} + (\beta_{\max} - \beta_{\min})\,\tfrac{e}{E}, \qquad &\text{(increasing)},\\ \mathrm{C}(e) &= \beta_{\max} - (\beta_{\max} - \beta_{\min})\,\tfrac{e}{E}, \qquad &\text{(decreasing)}. \end{aligned} \tag{19}$$

Where E refers to the total number of episodes. This linear curriculum offers a smooth and predictable transition between budget extremes. It is particularly useful in environments where safety tolerance should evolve steadily over time, for example, in robotic manipulation tasks where the agent gradually learns to apply larger forces after mastering safe low-force interactions.

Exponential (increasing / decreasing).

$$\begin{aligned} \mathrm{C}(e) &= \beta_{\min} + (\beta_{\max} - \beta_{\min})\big(1 - exp(-\kappa e)\big), && \text{(increasing)}, \\ \mathrm{C}(e) &= \beta_{\max} - (\beta_{\max} - \beta_{\min})\big(1 - exp(-\kappa e)\big), && \text{(decreasing)}, \end{aligned} \tag{20}$$

The exponential curriculum allows for rapid adaptation at the start of training while slowing the rate of change as the agent stabilizes. Such a schedule is well-suited for safety-critical navigation domains where initial fast exploration under tight constraints quickly transitions into steady, fine-grained learning of safe yet efficient behaviors.

Cosine (periodic increase–decrease).

$$\begin{aligned} \mathrm{C}(e) &= \bar{\beta} + \Delta\beta \cos\!\big(2\pi e/T\big), && \text{(increasing)}, \\ \mathrm{C}(e) &= \bar{\beta} - \Delta\beta \cos\!\big(2\pi e/T\big), && \text{(decreasing)}, \\ \bar{\beta} &= \tfrac{1}{2}(\beta_{\min} + \beta_{\max}),\ 0 < \Delta\beta \le \tfrac{1}{2}(\beta_{\max} - \beta_{\min}). \end{aligned} \tag{21}$$

This curriculum follows an cosine shaped increase or decrease based. It essentially acts a smoother version of the linear cirriulums and allows for more stable learning.

Across all curricula, our objective is to study whether the order and shape of budget progression influence learning speed, sample efficiency, and long-term adherence to safety constraints.

5.2 Budget-Aware Data Augmentation

A key challenge in BMDPs is that the Q-function must generalize across the continuous budget space $\mathcal{B}$, yet curriculum-based exploration collects data at sparse budget values. To address this, we introduce a simple data augmentation technique that generates multiple training samples from each transition by perturbing the budget parameter.

Method. For each collected transition $(s_i, \beta_i, a_i, \beta_i^a, r_i, c_i, s_i')$ at curriculum-scheduled budget $\beta_{\text{scheduled}}$, we generate N augmented versions with budgets sampled uniformly within a local neighborhood:

$$\beta_j^{\text{aug}} = \text{clip}\left(\beta_{\text{scheduled}} + \Delta\beta_j, 0, 1\right) \tag{22}$$

where $\Delta\beta_j \in [-\delta, +\delta]$ are uniformly spaced offsets. This transforms each transition into N augmented samples:

$$(s_i, \beta_i, a_i, \beta_i^a, r_i, c_i, s_i') \to \{(s_i, \beta_j^{\text{aug}}, a_i, \beta_i^a, r_i, c_i, s_i')\}_{j=1}^{N} \tag{23}$$

Implementation. We use $N = 7$ augmentations with radius $\delta = 0.12$, expanding the dataset from e.g. 50,000 to 350,000 transitions. This provides 5-10× more training data while maintaining curriculum alignment.

5.3 Temporal Granularity of Budget Updates

A key design choice concerns the temporal granularity of budget updates. In the per-episode setting, the initial budget β_0 is assigned at the beginning of each episode. This approach enforces a consistent constraint across the entire trajectory. In our work, we primarily adopt per-episode updates to remain consistent with the BMDP formulation and to promote stable learning. Note that in this context, the terms trajectory and episode are used interchangeably.

5.4 Evaluation Protocol

We evaluate the performance of budgeted reinforcement learning algorithms using three key metrics computed across multiple budget constraints (β values):

- **Overall Return:** The overall return measures the average performance across all budget levels. It is computed as the mean of the mean discounted rewards (R_d) obtained for each β value:

$$\text{Overall Return} = \frac{1}{|\mathcal{B}|}\sum_{\beta\in\mathcal{B}} \mathbb{E}[\mathcal{R}_\beta^\pi], \tag{24}$$

  where $\mathcal{B}$ is the set of all budget constraints tested, and $\mathbb{E}[\mathcal{R}_\beta^\pi]$ is the expected discounted return for policy π under budget constraint β. This metric captures the overall performance level achieved by the algorithm across different budget settings.
- **Mean Constraint Violation:** The mean constraint violation quantifies the extent to which the learned policy violates budget constraints. For each β value where the mean discounted cost exceeds β, we compute the violation amount. The metric is then the average violation across all violated budgets:

$$\text{Mean Violation} = \frac{1}{|\mathcal{V}|}\sum_{\beta\in\mathcal{V}} \max(0, \mathbb{E}[C_\beta^\pi] - \beta), \tag{25}$$

  where $\mathcal{V} = \{\beta \in \mathcal{B} \mid \mathbb{E}[C_\beta^\pi] > \beta\}$ is the set of violated budgets, and $\mathbb{E}[C_\beta^\pi]$ is the expected discounted cost under budget constraint β. A value of 0 indicates perfect constraint satisfaction.
- **Overall Cost:** The overall cost represents the average resource consumption across all budget levels. It is computed as the mean of the mean discounted costs (C_d) for each β value:

$$\text{Overall Cost} = \frac{1}{|\mathcal{B}|}\sum_{\beta\in\mathcal{B}} \mathbb{E}[C_\beta^\pi]. \tag{26}$$

  This metric provides insight into the resource efficiency of the learned policies across different budget constraints.

Experimental Setup. Network architectures and hyperparameters are kept constant across all curricula to ensure fair comparison. Policies are evaluated under both fixed and varying budget conditions to assess generalization and robustness.

5.5 Testing Environments

- **Autonomous driving**: The autonomous driving environment provides insight into how curriculum learning with varying initial beta values influences agent behaviour on a highway. The Highway-Env [13] is a simulated two-way-road where the agent is operating a vehicle. The road consists of two lanes: the right lane contains vehicles traveling in the same direction as the agent at slower speeds, while the left

Algorithm 1: Risk-sensitive exploration (BFTQ) with Curriculum Budget Initialization

```
Data: An environment $\mathcal{E}$, a BFTQ solver, $W$ CPU workers, curriculum $C$
Result: A batch of transitions $\mathcal{D}$
$\mathcal{D} \leftarrow \emptyset$
for each intermediate batch do
    split episodes between $W$ workers
    for each episode $e$ in batch in parallel do
        sample initial budget $\beta \leftarrow C(e)$          // Curriculum modification (replaces $\beta \sim \mathcal{U}(\mathcal{B})$)
        while episode not done do
            update $\varepsilon$ from schedule
            sample $z \sim \mathcal{U}([0,1])$
            if $z < \varepsilon$ then
                sample $(a, \beta_a) \sim \mathcal{U}(\Delta_{\mathcal{A}\times\mathcal{B}})$          // Explore
            else
                sample $(a, \beta_a) \sim \pi_{\text{greedy}}(a, \beta_a \mid s, \beta; Q^*)$          // Exploit
            append transition $(s, \beta, a, \beta_a, R, C, s')$ to batch $\mathcal{D}$
            step episode budget $\beta \leftarrow \beta_a$
    $\pi_{\text{greedy}}(\cdot \mid \cdot; Q^*) \leftarrow \text{BFTQ}(\mathcal{D})$
return $\mathcal{D}$
```

lane has oncoming traffic traveling in the opposite direction at higher speeds. The agent controls the car using a finite set of low-level actions: Idle(maintain speed and lane), Lane-left(change to left lane), Lane-right(change to right lane), Faster(accelerate), and Slower(decelerate). The reward function encourages high-velocity driving and lane positioning and the cost structure is the cumulative cost of the proportion of time spent on the opposite lane added to any collision penalty. This designs produces a clear risk-reward tradeoff with the incentive to overtake in the left lane when going behind a slower car and staying on the right lane when there is no car to overtake, balancing this against the safety cost of driving against incoming traffic.

- **Slot-Filling**: We tested our methodology also in the slot-filling dialogue environment originally proposed in Khouzaimi et al. [10], where an agent must complete a form by gathering user-provided information under automatic speech recognition (ASR) uncertainty, modeled through a fixed sentence error rate. At each step, the agent decides between faster but riskier oral inputs (`ask_oral`) and safer numeric inputs (`ask_num_pad`), balancing efficiency and reliability. Following Carrara et al. [6], we adopt this environment to analyze the safety–performance trade-off within the BMDP framework, using it to compare our proposed budget curricula against the BFTQ baseline under varying budget constraints.

6 Results

We evaluate curriculum learning across highway and slot-filling environments, comparing three schedule functions (linear, exponential, cosine) in both directions (strict→relaxed and relaxed→strict) against baseline BFTQ with uniform budget exploration [6].

6.1 Highway Environment

Table 2 shows that exponential S→R achieves the highest mean reward, improving approximately 5% over baseline. Cosine and linear R→S schedules also outperform baseline by 3-4%, while exponential R→S performs substantially worse, falling approximately 8% below baseline.

The baseline maintains the lowest constraint violations, while curriculum variants generally increase violation rates from 53% to 60%. However, Figure 2 reveals that different curriculum schedules dominate different budget regimes: exponential S→R excels in moderate-risk ranges (Figure 3c), while linear R→S shows competitive performance at specific budget values (Figure 2a).

Table 1 presents an ablation study showing that augmentation provides a cost-safety trade-off: slightly lower rewards but substantially reduced costs and constraint violations. Figure 1 visualizes how augmentation shifts policies toward safer regions of the reward-cost space. Given the main objective is to satisfy constraints, while maintaining high returns, it can be said that data augmentation is a net benefit.

6.2 Slot-Filling Environment

Table 3 demonstrates dramatically stronger curriculum benefits. Linear S→R achieves approximately 13% higher reward than baseline, the most significant improvement across all experiments. Exponential S→R matches baseline rewards while achieving the lowest costs and violations among all methods.

In contrast, all R→S schedules underperform baseline, with rewards 2-5% lower. This directional asymmetry is much stronger than in highway, suggesting curriculum effects are amplified in sequential decision tasks.

Figure 3 shows that linear S→R dominates the Pareto frontier for intermediate budgets (Figure 3a), achieving substantially higher rewards without cost increases. The baseline maintains the lowest

Table 1: Performance comparison between using data augmentation and not (Relaxed→Strict(Augmentation) and No Augmentation) in the highway environment.

Metric	Linear with Augmentation	Linear without Augmentation
Overall Return (mean R_d)	5.3871	**5.4241**
Overall Cost (mean C_d)	**0.3132**	0.3508
Mean Constraint Violation	**0.1308**	0.1681
Violation Rate	53.33%	53.33%

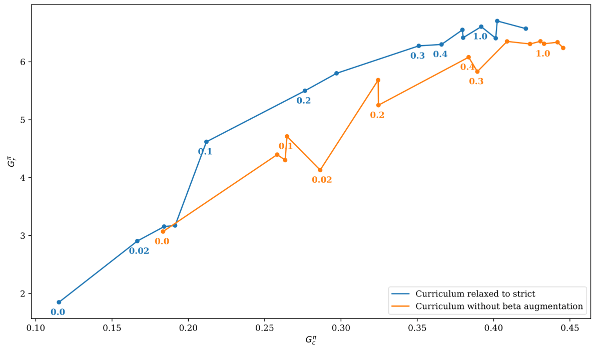
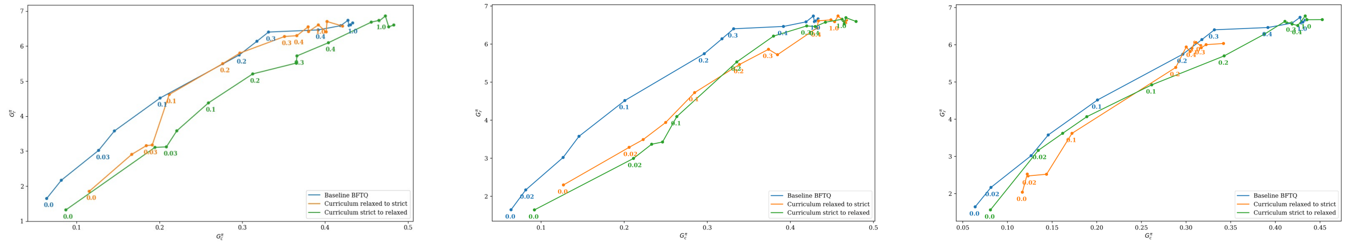


Figure 1: Comparison of reward-cost trade off when training without augmentation vs when training with.

Table 2: Performance comparison between baseline policy and different curriculum schedules (cosine, exponential, and linear) in the highway environment.

Metric	Baseline	Cosine R→S	Cosine S→R	Exp. R→S	Exp. S→R	Linear R→S	Linear S→R
Overall Return (mean R_d)	5.236	5.3981	5.3250	4.8202	**5.4888**	5.3871	5.2136
Overall Cost (mean C_d)	0.2993	0.3564	0.3593	**0.2531**	0.3310	0.3132	0.3491
Mean Constraint Violation	**0.077**	0.1410	0.1401	0.0797	0.1168	0.1308	0.1187
Violation Rate	53.33%	60.00%	60.00%	53.33%	60.00%	53.33%	60.00%



(a) Comparison between the baseline and linear curriculum.

(b) Comparison between the baseline and cosine curriculum.

(c) Comparison between the baseline and exponential curriculum.

Figure 2: Reward-cost trade-off plots for the highway environment.

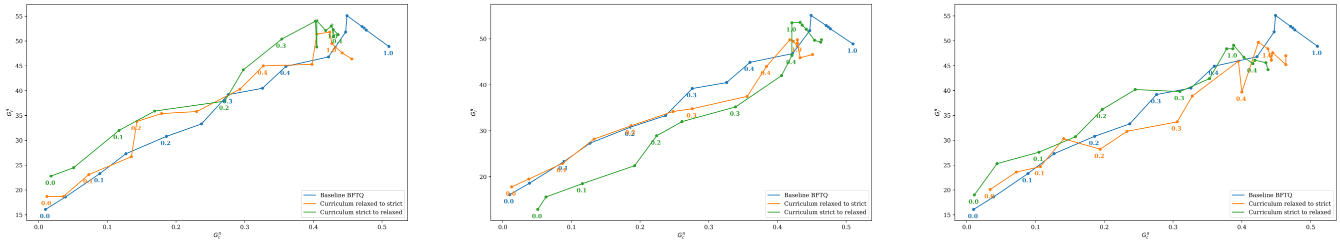
violation rate but is outperformed across most of the performance spectrum.

7 Discussion

A pattern emerges across both environments: strict→relaxed schedules generally outperform relaxed→strict schedules, often by substantial margins (Tables 2 and 3). This implies that learning safety before optimization produces superior outcomes. Starting with

Table 3: Performance comparison of baseline and curriculum-based policies (cosine, exponential and linear) in the slot filling environment.

Metric	Baseline	Cosine R→S	Cosine S→R	Exp. R→S	Exp. S→R	Linear R→S	Linear S→R
Overall Return (mean R_d)	39.6437	38.0687	38.4312	37.5562	39.6938	38.6437	**44.8375**
Overall Cost (mean C_d)	0.3063	0.2959	0.3241	0.3055	0.2929	**0.2758**	0.3160
Mean Constraint Violation	0.0100	0.0095	0.0287	0.0172	**0.0088**	0.0120	0.0354
Violation Rate	6.25%	12.50%	62.50%	25.00%	37.50%	6.25%	56.25%



(a) Comparison between the baseline and linear curriculum.

(b) Comparison between the baseline and cosine curriculum.

(c) Comparison between the baseline and exponential curriculum.

Figure 3: Reward-cost trade-off plots for the slot-filling environment

tight budgets forces agents to discover safe behaviors first, creating a foundation that transfers positively when budgets relax. Conversely, R→S schedules allow aggressive reward-seeking before safety awareness develops, requiring painful unlearning when constraints later tighten. The particularly poor exponential R→S performance in highway exemplifies this failure mode, where rapid early relaxation allows unsafe habits to become entrenched.

The magnitude of curriculum benefits varies dramatically between environments: slot-filling shows 2.7× larger improvements than highway. This difference reflects task structure: slot-filling's sequential dependencies, discrete action stages, and hierarchical objectives create natural curriculum structure that S→R schedules exploit effectively (Figure 3a).

Beyond direction, the optimal schedule shape varies by environment: exponential for highway, linear for slot-filling. Exponential schedules provide rapid initial learning then gradual refinement, matching highway's dynamics where safety fundamentals are quickly learned but overtaking strategies require careful tuning. Linear schedules provide steady progression matching slot-filling's staged dialogue structure. This alignment between schedule progression and task learning dynamics appears more important than mathematical properties like smoothness, as evidenced by cosine schedules' inconsistent results.

The ablation study (Table 1) reveals augmentation's role as safety regularization: slightly lower rewards but substantially improved constraint satisfaction. By training on multiple nearby budget values per transition, augmentation encourages smoother, more conservative Q-functions that generalize better across budget space, reducing violations from miscalibration. This trade-off typically favors augmentation in risk-averse applications.

However, curriculum methods face a fundamental dilemma: reward improvements often increase constraint violations compared to baseline. This arises because curriculum concentrates exploration along specific trajectories, improving performance in target regimes but potentially underfitting distant budget regions. Baseline uniform sampling explores all budgets equally, producing well-calibrated but conservative policies. Deployment strategy should match application requirements: use curriculum for performance-critical tasks accepting higher violations, exponential S→R or baseline for safety-critical applications, or multi-phase training combining both approaches.

8 Conclusion

This work proposed a budget curricula for BMDPs, delivering a structured approach to integrating curriculum learning within safe reinforcement learning. By systematically scheduling the evolution of initial budget values, we explored the influence of ordering and the progression of safety constraints on the learning process. Experiments across an autonomous driving and slot-filling environment demonstrated that curriculum design can meaningfully affect safety and performance outcomes.

Most effects were environment-dependent and not universally monotonic. Strict→relaxed performed better overall, but on the highway environment linear relaxed→strict performed best. The baseline BFTQ method remains a strong general-purpose baseline.

By unifying concepts from constrained reinforcement learning, BMDPs, and curriculum learning, this work advances the state of the art in safe reinforcement learning. Furthermore, it addreses a gap in literature by demonstrating how structured budget sequencing can be leveraged as a practical means to balance exploration efficiency with safety compliance.

Future work could extend this framework toward adaptive or performance-driven curricula, where the budget updates depend dynamically on agent behaviour rather than fixed schedules. Further validation could involve larger trajectory sets (e.g., increasing to $N_{\text{traj}} = 8000$).

References

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 2017. Constrained Policy Optimization. In *ICML (Proceedings of Machine Learning Research, Vol. 70)*. PMLR, 22–31.

[2] Eitan Altman. 1999. *Constrained Markov Decision Processes*. Chapman & Hall/CRC, Boca Raton.

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *ICML*. ACM, 41–48.

[4] Felix Berkenkamp, Matteo Turchetta, Angela P. Schoellig, and Andreas Krause. 2017. Safe Model-based Reinforcement Learning with Stability Guarantees. *CoRR* abs/1705.08551 (2017).

[5] Craig Boutilier and Tyler Lu. 2016. Budget allocation using weakly coupled, constrained Markov decision processes. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 52–61.

[6] Nicolas Carrara, Edouard Leurent, Romain Laroche, Tanguy Urvoy, Odalric-Ambrym Maillard, and Olivier Pietquin. 2019. Budgeted Reinforcement Learning in Continuous State Space. In *NeurIPS*. 9295–9305.

[7] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. 2015. Risk-Constrained Reinforcement Learning with Percentile Risk Criteria. *CoRR* abs/1512.01629 (2015).

[8] Javier García and Fernando Fernández. 2015. A comprehensive survey on safe reinforcement learning. *J. Mach. Learn. Res.* 16 (2015), 1437–1480.

[9] Sebastian Junges, Nils Jansen, Christian Dehnert, Ufuk Topcu, and Joost-Pieter Katoen. 2016. Safety-Constrained Reinforcement Learning for MDPs. In *TACAS*. Springer, 130–146.

[10] Hatim Khouzaimi, Romain Laroche, and Fabrice Lefèvre. 2015. Optimising Turn-Taking Strategies With Reinforcement Learning. In *SIGDIAL Conference*. The Association for Computer Linguistics, 315–324.

[11] Cevahir Köprülü, Thiago D. Simão, Nils Jansen, and Ufuk Topcu. 2025. Safety-Prioritizing Curricula for Constrained Reinforcement Learning. In *ICLR*. OpenReview.net.

[12] Ankita Kushwaha, Kiran Ravish, Preeti Lamba, and Pawan Kumar. 2025. A Survey of Safe Reinforcement Learning and Constrained MDPs: A Technical Survey on Single-Agent and Multi-Agent Safety. *CoRR* abs/2505.17342 (2025).

[13] Edouard Leurent. 2018. An Environment for Autonomous Driving Decision-Making. https://github.com/eleurent/highway-env.

[14] Yongshuai Liu, Jiaxin Ding, and Xin Liu. 2019. IPO: Interior-point Policy Optimization under Constraints. *CoRR* abs/1910.09615 (2019).

[15] Zuxin Liu, Zijian Guo, Yihang Yao, Zhepeng Cen, Wenhao Yu, Tingnan Zhang, and Ding Zhao. 2023. Constrained Decision Transformer for Offline Safe Reinforcement Learning. In *ICML*. PMLR, 21611–21630.

[16] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. 2020. Curriculum Learning for Reinforcement Learning Domains: A Framework and Survey. *J. Mach. Learn. Res.* 21 (2020), 181:1–181:50.

[17] Rémy Portelas, Cédric Colas, Lilian Weng, Katja Hofmann, and Pierre-Yves Oudeyer. 2020. Automatic Curriculum Learning For Deep RL: A Short Survey. In *IJCAI*. ijcai.org, 4819–4825.

[18] Julien Roy, Roger Girgis, Joshua Romoff, Pierre-Luc Bacon, and Christopher J. Pal. 2022. Direct Behavior Specification via Constrained Reinforcement Learning. In *ICML*. PMLR, 18828–18843.

[19] Shahaf S. Shperberg, Bo Liu, and Peter Stone. 2024. Relaxed Exploration Constrained Reinforcement Learning. In *AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems / ACM, 1727–1735.

[20] Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. 2018. Reward Constrained Policy Optimization. *CoRR* abs/1805.11074 (2018).

[21] Matteo Turchetta, Andrey Kolobov, Shital Shah, Andreas Krause, and Alekh Agarwal. 2020. Safe Reinforcement Learning via Curriculum Induction. In *NeurIPS*.

[22] Akifumi Wachi, Xun Shen, and Yanan Sui. 2024. A Survey of Constraint Formulations in Safe Reinforcement Learning. In *IJCAI*. ijcai.org, 8262–8271.

[23] Qisong Yang. 2023. *Risk Aversion and Guided Exploration in Safety-Constrained Reinforcement Learning*. Dissertation (TU Delft). Delft University of Technology. doi:10.4233/uuid:ca5a81c2-f895-4638-bce5-1423a5943381

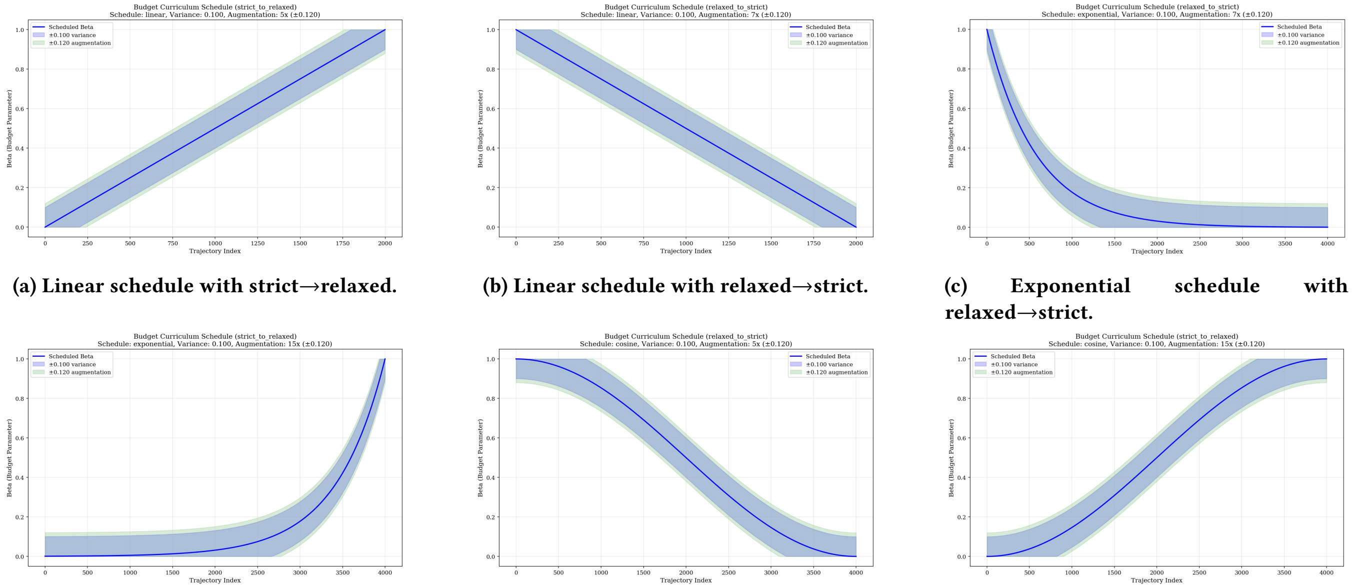
Appendix

A Notation

Basic symbols.

$\mathcal{S}, \mathcal{A}$	State and action spaces
$P(s' \mid s, a)$	Transition kernel
$R_r(s,a)$, $R_c(s,a)$	Immediate reward and cost
$\gamma \in [0,1)$	Discount factor
$\beta \in \mathcal{B} \subset \mathbb{R}_+$	Budget (constraint threshold)
$\tilde{s} = (s, \beta)$, $\tilde{a} = (a, \beta_a)$	Augmented state and action
$\tilde{\mathcal{S}} = \mathcal{S} \times \mathcal{B}$, $\tilde{\mathcal{A}} = \mathcal{A} \times \mathcal{B}$	Augmented spaces
G_r^π, G_c^π	Discounted returns of reward and cost
$Q = (Q_r, Q_c)$	Vector action-value function
$e \in \mathbb{N}$	Episode index ($\mathbb{N}$: natural numbers)
$\mathrm{C} : \mathbb{N} \to \mathcal{B}$	Curriculum schedule for initial budgets
$\beta_{\min}$, $\beta_{\max}$	Minimum and maximum budgets

B Budget Curriculum Schedules



(a) Linear schedule with strict→relaxed.

(b) Linear schedule with relaxed→strict.

(c) Exponential schedule with relaxed→strict.

(d) Exponential schedule with strict→relaxed.

(e) Cosine schedule with relaxed→strict.

(f) Cosine schedule with relaxed→strict.

Figure 4: All panels depict the scheduled β_0 trajectory (line) over training; higher β_0 relaxes safety constraints, expanding the feasible action space.