RBGA: RESIDUAL-BUDGET GUIDE ARBITRATION FOR OFFLINE-TO-ONLINE SAFE REINFORCEMENT LEARNING IN DEXTEROUS MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Dexterous manipulation requires robots to leverage existing datasets for rapid skill acquisition while maintaining safety during real-world deployment to prevent damage to the robot or its surroundings. Offline-to-online safe reinforcement learning (O2O safe RL) provides a promising solution for this problem by leveraging prior experience for online improvement under safety constraints, thereby mitigating the sample inefficiency and exploration risk of learning safe policies from scratch. However, existing O2O safe RL approaches do not explicitly account for the safety budget already consumed when making decisions within an episode. Excessive cost incurred early can therefore leave little room for subsequent exploration and policy improvement. To address this limitation, we propose **R**esidual-**B**udget **G**uide **A**rbitration (RBGA), an algorithm that explicitly tracks the remaining episode safety budget and uses it to adaptively select between offline guidance and the evolving online policy. At each step, RBGA compares the cost-to-go estimates of the guide and online actions against the remaining budget, enabling step-wise action arbitration throughout the episode. We provide theoretical analysis characterizing the safety implications of residual-budget-based selection and its effect on guide-assisted exploration. Extensive experiments demonstrate that RBGA enables effective online policy improvement while maintaining strong safety performance.

1 INTRODUCTION

Dexterous manipulation enables multi-fingered robotic hands to perform a wide range of contact-rich tasks, including in-hand manipulation, grasping, and tool use (Rajeswaran et al., 2018; OpenAI et al., 2020; Yuan et al., 2025). Despite substantial progress, improving the safety of dexterous manipulation remains a pressing challenge (García & Fernández, 2015; Jain et al., 2023; Geng et al., 2025). Safe reinforcement learning (RL) addresses this challenge by optimizing task performance subject to explicit cost constraints (Chow et al., 2018; Tessler et al., 2019; Stooke et al., 2020; Satija et al., 2020). However, learning a safe dexterous policy from scratch is often prohibitive, as it requires extensive environment interactions that are both costly and risky (Ray et al., 2019; Thananjeyan et al., 2021; Dulac-Arnold et al., 2021). Offline safe RL reduces this burden by learning policies from a fixed dataset without additional online interaction (Xu et al., 2022; Lee et al., 2022a; Liu et al., 2023). Yet its performance is constrained by the coverage of the offline data (Levine et al., 2020; Liu et al., 2024), making learned policies vulnerable to distribution shift and out-of-distribution (OOD) actions when encountering unseen states (Fujimoto et al., 2019; Kumar et al., 2019; 2020). Offline-to-online (O2O) safe RL therefore offers an appealing alternative by leveraging policies learned offline to accelerate online adaptation, substantially reducing the extensive environment interactions required to learn policies from scratch.

Compared with O2O RL, O2O safe RL remains relatively underexplored (Li et al., 2024; Cao et al., 2025; Chen et al., 2026). Existing policy-finetuning methods continue online learning from an offline-pretrained policy, although this transfer can suffer from mismatched value estimates and Lagrange multipliers between offline and online learning (Chen et al., 2026). To facilitate safer and more effective adaptation, some works instead use an offline policy to guide the learning of a separate online policy (Li et al., 2024), while others preserve offline knowledge through model-based fine-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tuning and safe policy expansion (Cao et al., 2025). Despite these advances, existing methods do not explicitly account for the cumulative cost already incurred in the current episode when making online decisions. This limitation is particularly important in dexterous manipulation, where contact-rich interactions can produce abrupt safety violations and nonuniform cost accumulation over an episode. Therefore, developing a dedicated O2O safe RL algorithm that explicitly accounts for within-episode cost accumulation is particularly important for dexterous manipulation.

In this work, to address the lack of trajectory-level budget awareness in O2O safe RL, we propose Residual-Budget Guide Arbitration (RBGA), an algorithm that explicitly incorporates the remaining episode cost budget into online policy learning. First, RBGA tracks the remaining cost budget along each trajectory to characterize the current safety margin throughout online interaction. Then, at each interaction step, it evaluates the actions proposed by the offline guide and the online policy based on their estimated cost-to-go, and adaptively selects between them according to the remaining budget. This step-wise arbitration allows the online policy to take control when the guide action becomes unfavorable under the current budget, while retaining useful offline guidance when it remains admissible. Finally, extensive experiments on Redman (Geng et al., 2025) demonstrate the effectiveness of RBGA across a range of dexterous manipulation tasks. The main contributions of this work are summarized as follows:

- To the best of our knowledge, we are the first to explicitly incorporate trajectory-level episode cost budget information into O2O safe RL for dexterous manipulation.
- We propose RBGA, a step-wise action arbitration algorithm that adaptively balances offline guidance and the evolving online policy based on the remaining episode budget and estimated cost-to-go.
- Extensive experiments across diverse dexterous manipulation tasks demonstrate the effectiveness and advantages of RBGA, showing strong policy performance while maintaining favorable safety behavior during online adaptation.

2 Related Work

RL for Dexterous Manipulation. RL has enabled multi-fingered robotic hands to acquire complex manipulation skills from interaction, with early studies demonstrating effective learning from demonstrations and large-scale simulation (Rajeswaran et al., 2018; Zhu et al., 2019; OpenAI et al., 2020). Subsequent work has improved the applicability of learned policies through model-based planning and more efficient policy optimization (Nagabandi et al., 2020; Charlesworth & Montana, 2021). Recent studies further extend dexterous RL to unseen objects and real-world manipulation (Chen et al., 2022a; Qin et al., 2023; Qi et al., 2023a;b), as well as more challenging bimanual tasks (Chen et al., 2022b; Huang et al., 2023). Despite these advances, most prior work considers dexterous manipulation without explicit safety constraints. Safety is nevertheless critical for dexterous manipulation, as unsafe actions during contact-rich interaction may damage the robotic hand or its surroundings. Research on safe dexterous manipulation remains limited. A few studies incorporate explicit constraints into dexterous policy learning or extend safe RL to bimanual manipulation (Jain et al., 2023; Zhan & Chin, 2024). ReDMan further establishes a dedicated benchmark for evaluating safe RL algorithms on dexterous manipulation tasks (Geng et al., 2025). These works mainly learn or evaluate safe policies through online interaction, leaving how offline experience can facilitate subsequent safe adaptation largely unexplored.

Offline-to-Online RL. Offline-to-online (O2O) RL leverages offline pretraining to accelerate online policy improvement. A natural strategy is to initialize online learning with an offline-trained policy and continue fine-tuning it using collected experience (Nair et al., 2020). To improve the stability and efficiency of this transition, some methods balance offline and online samples during replay (Lee et al., 2022b; Ball et al., 2023), while others correct value estimates inherited from offline training before or during online fine-tuning (Yu & Zhang, 2023; Nakamoto et al., 2023; Zhang et al., 2024; Luo et al., 2024). Instead of directly fine-tuning the offline policy, JSRL uses a fixed guide policy to facilitate exploration through progressively shortened roll-ins (Uchendu et al., 2023), whereas PEX retains separate offline and online policies and selects between their proposed actions according to estimated values (Zhang et al., 2023). Despite substantial progress, most O2O RL methods are developed for unconstrained reward maximization and do not address safety constraints during online adaptation. Recent O2O safe RL studies begin to address this problem by incorporating safety into

offline-to-online transfer. GOLD uses an offline sequence policy to guide the learning of an online safe policy (Li et al., 2024). However, GOLD relies on a Decision Transformer as the offline guide and performs guided online distillation, differing from the matched offline-policy setting considered in this work. FOSP combines offline model learning with safe policy expansion for online fine-tuning (Cao et al., 2025). However, FOSP adopts a model-based safe O2O setting with a learned world model, which is not directly comparable to the model-free setting considered in this work.

Despite these advances, existing O2O safe RL studies have not addressed dexterous manipulation, where unsafe exploratory actions can damage the robotic hand or its surroundings. Learning a safe dexterous policy from scratch requires costly and risky online interaction, while offline safe RL remains limited by the coverage of the fixed dataset and degrades under distribution shift. O2O safe RL is therefore particularly appealing for dexterous manipulation, as it reuses offline knowledge to accelerate online adaptation under safety constraints. To the best of our knowledge, we are the first to study O2O safe RL for dexterous manipulation.

3 PRELIMINARIES

This section introduces the constrained RL formulation and the O2O safe RL setting used throughout the paper. We first describe finite-horizon Markov decision processes and their extension to explicit cost constraints. We then specify the offline and online policies considered by RBGA.

3.1 CONSTRAINED MARKOV DECISION PROCESSES

Standard RL is formulated as a finite-horizon Markov decision process (MDP) $\mathcal{M}_{\mathrm{MDP}} = (\mathcal{S}, \mathcal{A}, P, r, \rho_0, \gamma, H)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, $P(\cdot \mid s, a)$ is the transition kernel, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a bounded reward function, ρ_0 is the initial-state distribution, $\gamma \in (0, 1]$ is the discount factor, and H is the finite episode horizon. A policy π induces a trajectory through $s_0 \sim \rho_0$, $a_t \sim \pi(\cdot \mid s_t)$, and $s_{t+1} \sim P(\cdot \mid s_t, a_t)$. For policies that depend on the episode step, we omit the time index for simplicity. Standard RL maximizes the expected discounted reward.

Safe RL augments the MDP with a bounded non-negative cost function $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and a cost limit d, giving the constrained MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, c, \rho_0, \gamma, H, d)$ (Altman, 1999; Achiam et al., 2017). For any policy π, the expected reward and cost are

$$J_R(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t)\right], \qquad J_C(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{H-1} \gamma^t c(s_t, a_t)\right], \tag{1}$$

where τ denotes a trajectory induced by policy π, and R and C denote reward and cost, respectively.

The common objective of constrained RL is to maximize the cumulative reward while satisfying the cumulative cost constraint. Accordingly, the optimal feasible policy $\pi^\star$ is defined as

$$\pi^\star \in \arg\max_{\pi} J_R(\pi) \quad \text{s.t.} \quad J_C(\pi) \le d. \tag{2}$$

The objective maximizes task return while constraining the expected cumulative cost. In dexterous manipulation, the state represents the hand–object configuration, the action specifies a control command for the robotic hand, and the reward measures progress toward the manipulation objective, while the cost function penalizes unsafe behavior during interaction.

3.2 OFFLINE-TO-ONLINE SAFE REINFORCEMENT LEARNING

Offline-to-online (O2O) RL leverages offline data or a previously learned policy to support subsequent online policy learning (Uchendu et al., 2023; Zhang et al., 2023), while O2O safe RL further requires this online improvement to satisfy the cost constraint in Eq. (2) (Li et al., 2024; Cao et al., 2025; Chen et al., 2026). In O2O safe RL, an offline policy π_g is first trained on a fixed offline dataset $\mathcal{D}_{\mathrm{off}}$ of transitions (s, a, r, c, s', δ), where $r = r(s, a)$ and $c = c(s, a)$ are the observed reward and cost, s' is the next state, and $\delta \in \{0, 1\}$ indicates episode termination. The learned offline policy is then used to support subsequent online policy learning. We denote the online policy by π_o and update it from newly collected transitions using a constrained RL algorithm.

4 METHODOLOGY

In this section, we first introduce Residual-Budget Guide Arbitration (RBGA) for O2O safe RL in dexterous manipulation. We first describe how guide horizon scheduling and residual-budget-based action selection combine offline guidance with online learning. We then provide theoretical analysis of the cost of the composite policy and the reward performance of the learned online policy.

4.1 RESIDUAL-BUDGET GUIDE ARBITRATION

Guide horizon scheduling. Inspired by JSRL (Uchendu et al., 2023), RBGA combines a fixed offline guide policy π_g and a trainable online policy π_o into a composite policy π for online interaction. Consider a dexterous manipulation episode of at most H control steps and a total of N online training iterations. To limit dependence on the offline guide, we specify a guidance cutoff $N_g \leq N$. During the first N_g training iterations, the offline guide participates in action selection, while the online policy acts alone thereafter. The online policy is initialized independently and updated from collected experience. At iteration i, the maximum guide horizon $H_i \in \{0, \dots, H\}$ controls the number of initial episode steps during which guide actions will be considered. Decreasing H_i allows the online policy to take control earlier as training progresses. Specifically, in JSRL, the guide horizon is directly reduced across curriculum stages, introducing abrupt changes in the state distribution encountered by the online policy. To provide a smoother transition from offline guidance to online control, we instead gradually decrease the maximum guide horizon as training progresses and consider two scheduling strategies, *Linear* and *Step*. For the linear schedule, we set

$$H_i = \mathrm{round}\left(H_0 \left[1 - \frac{i}{N_g} \right]_+ \right), \qquad N_g > 0, \tag{3}$$

where $i \in \{0, \dots, N\}$ counts completed online training iterations, $H_0 \leq H$ is the initial maximum guide horizon, $[\cdot]_+$ clips negative values to zero, and $\mathrm{round}(\cdot)$ rounds to the nearest integer. Thus, $H_i = 0$ for $i \geq N_g$. A step schedule is described in Appendix A.5.

To allow the online policy to take over at different stages of an episode, we sample h uniformly from $\{0, \dots, H_i\}$ at the beginning of each episode, independently of the initial state. The sampled horizon h remains fixed within that episode, and guide actions are considered only when $t < \min\{h, H_i\}$. Here, t counts control steps within the current manipulation attempt, while i counts training iterations. If an episode spans several updates, the current H_i continues to apply, so a decrease in H_i can end guidance before the sampled horizon h is reached.

Residual budget. A limited guide horizon does not by itself prevent unsafe actions. The offline guide may produce actions that substantially violate safety constraints and potentially damage the robotic hand or its surroundings due to distribution shift between offline training and online interaction. We therefore account for the costs already incurred when selecting actions within the guide horizon. Before executing the action at step t, we define the residual budget as

$$b_t = d - \sum_{j=0}^{t-1} \gamma^j c_j, \qquad b_0 = d, \tag{4}$$

where d is the cost limit for a complete manipulation episode, $c_j = c(s_j, a_j)$ measures the safety cost incurred by the hand at step j, and γ is the discount factor. In dexterous tasks, the cost penalizes unsafe behaviors of the robotic hand during manipulation. Thus, b_t is the budget left for the rest of the manipulation, and $b_t < 0$ means that the accumulated discounted cost has already exceeded d.

To select an action under the residual budget, we first obtain candidate actions from the two policies. Given the current state s_t, we sample an online candidate $a_t^o \sim \pi_o(\cdot \mid s_t)$ and, when $t < \min\{h, H_i\}$, a guide candidate $a_t^g \sim \pi_g(\cdot \mid s_t)$. To compare the candidates against the residual budget, we evaluate the expected discounted cost of executing each candidate and following the current online policy thereafter. For a fixed π_o, this cost-to-go is defined as:

$$Q_C^{\pi_o}(s_t, a_t) = \mathbb{E}_{\pi_o}\left[\sum_{j=t}^{H-1} \gamma^{j-t} c_j \,\middle|\, s_t, a_t \right], \tag{5}$$

Algorithm 1 Residual-Budget Guide Arbitration (RBGA)

Input: fixed guide policy π_g, task horizon H, total training iterations N, guide horizon schedule with cutoff N_g, cost limit d, discount factor γ, and comparison margin ζ.
Initialize the online learner with policy π_o and the auxiliary estimator $\widehat{Q}_C^{\pi_o}$ independently of π_g; set $\mathcal{B} \leftarrow \varnothing$ and $i \leftarrow 0$.
Reset the environment; set $t \leftarrow 0$, $b_t \leftarrow d$, and sample $h \sim \mathrm{Unif}\{0, \ldots, H_i\}$.
while $i < N$ **do**
  for each environment step **do**
    Select a_t using Eq. (6) after warm-up; use the warm-up procedure otherwise.
    Execute a_t; observe $r_t, c_t, s_{t+1}, \delta_t$; append the transition to $\mathcal{B}$.
    Set $b_{t+1} \leftarrow b_t - \gamma^t c_t$ and $t \leftarrow t + 1$.
    if the episode has ended **then**
      Reset the environment; set $t \leftarrow 0$, $b_t \leftarrow d$, and sample a new $h \sim \mathrm{Unif}\{0, \ldots, H_i\}$.
    end if
  end for
  $\pi_o, \widehat{Q}_C^{\pi_o}, i \leftarrow \textsc{TrainPolicy}(\pi_o, \widehat{Q}_C^{\pi_o}, \mathcal{B}, i, N)$; obtain H_i from the schedule.
end while
Return: the learned online policy π_o.

where a_t is the candidate control command being evaluated and subsequent actions follow π_o. For notational simplicity, we omit the time index on finite-horizon value and advantage functions when the corresponding episode step is clear from context. Since this expected cost is not directly available, we learn an auxiliary cost estimator $\widehat{Q}_C^{\pi_o}(s, a)$ from online experience to approximate Eq. (5) and use it to evaluate both candidates, as detailed in Appendix A.4. For comparison with b_t, we define $\widehat{q}_t^{\,g} = \gamma^t \widehat{Q}_C^{\pi_o}(s_t, a_t^g)$ and $\widehat{q}_t^{\,o} = \gamma^t \widehat{Q}_C^{\pi_o}(s_t, a_t^o)$. These estimates express the candidates' remaining costs on the same discount scale as the episode budget.

Step-wise arbitration. RBGA combines the guide horizon, residual budget, and estimated cost-to-go to select the executed action. To retain useful guidance while limiting its additional cost, we introduce a comparison margin $\zeta \geq 0$, which bounds the additional estimated cost-to-go of the guide candidate relative to the online candidate. The executed action is then selected as

$$a_t = \begin{cases} a_t^o, & t \geq \min\{h, H_i\}, \\ a_t^g, & t < \min\{h, H_i\} \text{ and } \widehat{q}_t^{\,g} \leq \min\{b_t, \widehat{q}_t^{\,o} + \gamma^t \zeta\}, \\ a_t^o, & t < \min\{h, H_i\},\ \widehat{q}_t^{\,o} \leq b_t \text{ and } \widehat{q}_t^{\,g} > \min\{b_t, \widehat{q}_t^{\,o} + \gamma^t \zeta\}, \\ \arg\min_{a \in \{a_t^g, a_t^o\}} \widehat{Q}_C^{\pi_o}(s_t, a), & \text{otherwise.} \end{cases} \tag{6}$$

When $t \geq \min\{h, H_i\}$, the guide is no longer eligible, and RBGA executes the online candidate a_t^o. When $t < \min\{h, H_i\}$ and $\widehat{q}_t^{\,g} \leq \min\{b_t, \widehat{q}_t^{\,o} + \gamma^t \zeta\}$, the guide remains eligible, its estimated cost-to-go satisfies the remaining budget, and its additional estimated cost relative to the online candidate is within the comparison margin; RBGA therefore selects a_t^g. When $t < \min\{h, H_i\}$, $\widehat{q}_t^{\,o} \leq b_t$, and $\widehat{q}_t^{\,g} > \min\{b_t, \widehat{q}_t^{\,o} + \gamma^t \zeta\}$, the online candidate satisfies the remaining budget while the guide fails either the budget or comparison-margin condition, so RBGA selects a_t^o. Otherwise, both candidates exceed the remaining budget, and RBGA selects the candidate with the smaller estimated cost-to-go. We provide a detailed description of RBGA in Algorithm 1.

4.2 Theoretical Analysis

In this section, we analyze the cost of RBGA action selection and the reward performance of the learned policy. We first derive local cost bounds from the remaining budget and cost estimates, and then obtain an expected cost bound for the composite policy. Subsequently, we analyze the reward gap relative to the optimal feasible policy under state coverage and policy advantage conditions. To account for the accuracy of the cost estimates in Eq. (6), we introduce the following assumption.

Assumption 4.1 (Cost estimation error)**.** *For a fixed online policy π_o and cost estimator $\widehat{Q}_C^{\pi_o}$, there exists a finite constant $\varepsilon_C \geq 0$ such that, at each time step $t < \min\{h, H_i\}$, the cost estimates of*

both candidate actions in Eq. (6) satisfy:

$$\left|\widehat{Q}_C^{\pi_o}(s_t,a) - Q_C^{\pi_o}(s_t,a)\right| \le \varepsilon_C, \qquad \forall a \in \{a_t^g, a_t^o\}, \tag{7}$$

where ε_C is a uniform bound on their estimation errors.

Based on Assumption 4.1 and the remaining budget in Eq. (4), we derive the cost bounds for a selected action, as shown in Proposition 4.2. Proofs and discussions are in Appendix C.2.

Proposition 4.2 (Local cost bounds)**.** *Under Assumption 4.1, consider a time step $t < \min\{h, H_i\}$. For a selected action a_t satisfying $\gamma^t \widehat{Q}_C^{\pi_o}(s_t,a_t) \le b_t$, the accumulated discounted cost plus the expected discounted cost of executing a_t and following π_o thereafter satisfies:*

$$\sum_{j=0}^{t-1} \gamma^j c_j + \gamma^t Q_C^{\pi_o}(s_t,a_t) \le d + \gamma^t \varepsilon_C \le d + \varepsilon_C. \tag{8}$$

For every guide action selected by Eq. (6), the difference between its cost-to-go and that of the online candidate satisfies:

$$Q_C^{\pi_o}(s_t,a_t^g) - Q_C^{\pi_o}(s_t,a_t^o) \le \zeta + 2\varepsilon_C. \tag{9}$$

Proposition 4.2 bounds the accumulated discounted cost together with the expected continuation cost for actions satisfying the budget condition, and the additional cost-to-go of a guide action relative to the online candidate. By combining Eq. (9) with the finite-horizon cost form of the performance difference lemma (Kakade & Langford, 2002), we obtain the episode cost bound in Proposition 4.3.

Proposition 4.3 (Cost bound for the composite policy)**.** *Under Assumption 4.1, fix π_o, $\widehat{Q}_C^{\pi_o}$, and H_i throughout an episode, and sample $h \sim \mathrm{Unif}\{0,\dots,H_i\}$ independently of the initial state. The composite policy π defined by Eq. (6) satisfies:*

$$[J_C(\pi) - d]_+ \le [J_C(\pi_o) - d]_+ + (\zeta + 2\varepsilon_C) \sum_{t=0}^{H_i-1} \gamma^t \frac{H_i - t}{H_i + 1}, \tag{10}$$

where $[u]_+ = \max\{u, 0\}$, $J_C(\pi)$ is the expected discounted episode cost under the composite policy, with expectation over the initial state, sampled guide horizon, candidate actions, and state transitions, and $(H_i - t)/(H_i + 1) = \Pr(h > t)$ is the probability that the sampled guide horizon extends beyond step t under uniform sampling, for $0 \le t < H_i$.

Proposition 4.3 bounds the additional violation of the expected cost constraint in terms of the comparison margin, estimation error, and maximum guide horizon. The final term does not increase as the maximum guide horizon is reduced and equals zero when the guide is removed. Proposition C.1 in Appendix C.3 extends the analysis to policy updates within an episode.

To analyze reward performance under RBGA roll-ins, we consider the state distributions used for policy learning, following the state coverage analysis of guide-based policy search (Uchendu et al., 2023). For a learned online policy π_o, the reward advantage at time t is defined as:

$$A_R^{\pi_o}(s,a) = Q_R^{\pi_o}(s,a) - \mathbb{E}_{a' \sim \pi_o(\cdot|s)}[Q_R^{\pi_o}(s,a')], \tag{11}$$

where $Q_R^{\pi_o}(s,a)$ is the expected discounted reward $\sum_{j=t}^{H-1} \gamma^{j-t} r_j$ obtained by executing a in s and following π_o thereafter, and $r_j = r(s_j, a_j)$. Based on this advantage function, we introduce conditions on state coverage and the remaining advantage of the optimal feasible policy.

Assumption 4.4 (State coverage and policy advantage)**.** *For the learned online policy π_o, there exist a finite coverage coefficient $D_\infty \ge 1$ and finite reward advantage bounds $\varepsilon_{R,t} \ge 0$ such that, at every time step t, the state distributions satisfy:*

$$\rho_t^{\pi^\star} \le D_\infty \mu_t, \tag{12}$$

where μ_t is the marginal state distribution at time t induced by RBGA roll-ins, $\rho_t^{\pi^\star}$ is the corresponding distribution under $\pi^\star$, and D_∞ bounds their relative probabilities. Under the roll-in distribution, the expected positive advantage of the optimal feasible policy satisfies:

$$\mathbb{E}_{s \sim \mu_t}\left[\left[\mathbb{E}_{a \sim \pi^\star(\cdot|s)} A_R^{\pi_o}(s,a)\right]_+\right] \le \varepsilon_{R,t}, \tag{13}$$

where $\varepsilon_{R,t}$ bounds the remaining positive reward advantage of $\pi^\star$ under the roll-in distribution.

By applying the state coverage condition to the positive advantage in Eq. (13), we obtain the reward performance bound shown in Proposition 4.5. Proofs and discussions are in Appendix C.4 and C.5.

Proposition 4.5 (Reward performance)**.** *Under Assumption 4.4, the reward gap between the learned online policy π_o and the optimal feasible policy $\pi^\star$ satisfies:*

$$J_R(\pi^\star) - J_R(\pi_o) \leq D_\infty \sum_{t=0}^{H-1} \gamma^t \varepsilon_{R,t}. \tag{14}$$

Proposition 4.5 bounds the reward gap between the learned online policy π_o and the optimal feasible policy $\pi^\star$. Once $H_i = 0$, Eq. (6) gives $\pi = \pi_o$. The same bound therefore applies to the composite policy after the guide is removed, provided that the learned online policy satisfies Assumption 4.4.

5 EXPERIMENT

In this section, we conduct comprehensive comparative experiments between RBGA and prior O2O safe RL methods in dexterous manipulation tasks. Moreover, we conduct ablation studies to evaluate the effectiveness of guide horizon scheduling and the contribution of residual-budget arbitration.

5.1 TASK AND BASELINE

Task. We evaluate RBGA on four dexterous manipulation tasks from the ReDMan benchmark (Geng et al., 2025), including *Joint*, *Finger*, *ArmJoint*, and *ArmFinger*. These tasks combine two bimanual object-transfer settings with constraints on either a single joint or multiple finger joints; task descriptions are provided in Appendix A.1. The original ReDMan environments use binary cost signals to indicate safety violations, resulting in abrupt changes between safe and unsafe states. To provide a more informative safety signal for policy learning, we replace the binary cost with a continuous cost while preserving the original safety definition of each task. The resulting continuous cost signal $c_t = c(s_t, a_t)$ is used to compute the cumulative cost over an episode. We set the episode cost limit to $d = 30$ for all tasks.

Baselines. We compare RBGA with the following six baselines: *(i)* **JSRL-Random-Lag** (Uchendu et al., 2023): the random-switching variant of JSRL, which uniformly samples the number of initial guide steps for each episode and then switches from the offline guide to the online policy; *(ii)* **JSRL-Curr-Lag** (Uchendu et al., 2023): the curriculum variant of JSRL, which progressively reduces the number of initial guide steps during training, allowing the online policy to take control earlier in each episode; *(iii)* **PEX-Lag** (Zhang et al., 2023): adaptively selects between actions proposed by a fixed offline policy and a trainable online policy using reward value estimates with a Lagrangian cost penalty; *(iv)* **SO2-Lag** (Zhang et al., 2024): fine-tunes an offline-pretrained policy using a mixture of offline and online replay data, perturbed target actions, and more frequent critic updates; *(v)* **Direct**: initializes the online actor and the reward and cost critics from an offline-pretrained checkpoint, and then directly fine-tunes them using SAC-Lag; *(vi)* **Scratch**: trains SAC-Lag from scratch using online interactions, without offline pretraining. Since JSRL, PEX, and SO2 were originally developed for unconstrained O2O RL, we adapt them to O2O safe RL by using cost critics and adaptive Lagrange multipliers to incorporate episode cost constraints into online policy learning, following the Lagrangian formulation used in RBGA. We additionally report **Offline**, the performance of the offline-pretrained policy without online fine-tuning, as a reference.

Implementation Details. We use IQL as the offline policy and SAC-Lag as the online policy. See the Appendix A for implementation details. Unless otherwise stated, RBGA uses the Linear schedule with $N_g = 40{,}000$ iterations. In addition, we evaluate the compatibility of RBGA with DDPG-Lag (Lillicrap et al., 2016; Ji et al., 2024) as an alternative online learner, with detailed results provided in the Appendix B.2.

5.2 PERFORMANCE COMPARISON EXPERIMENT

To evaluate the performance of RBGA across various dexterous manipulation tasks, we conduct comprehensive comparative experiments. The results involve comparing RBGA with the baseline algorithms on four tasks: *Finger*, *Joint*, *ArmFinger*, and *ArmJoint*. Figure 1 illustrates the reward,

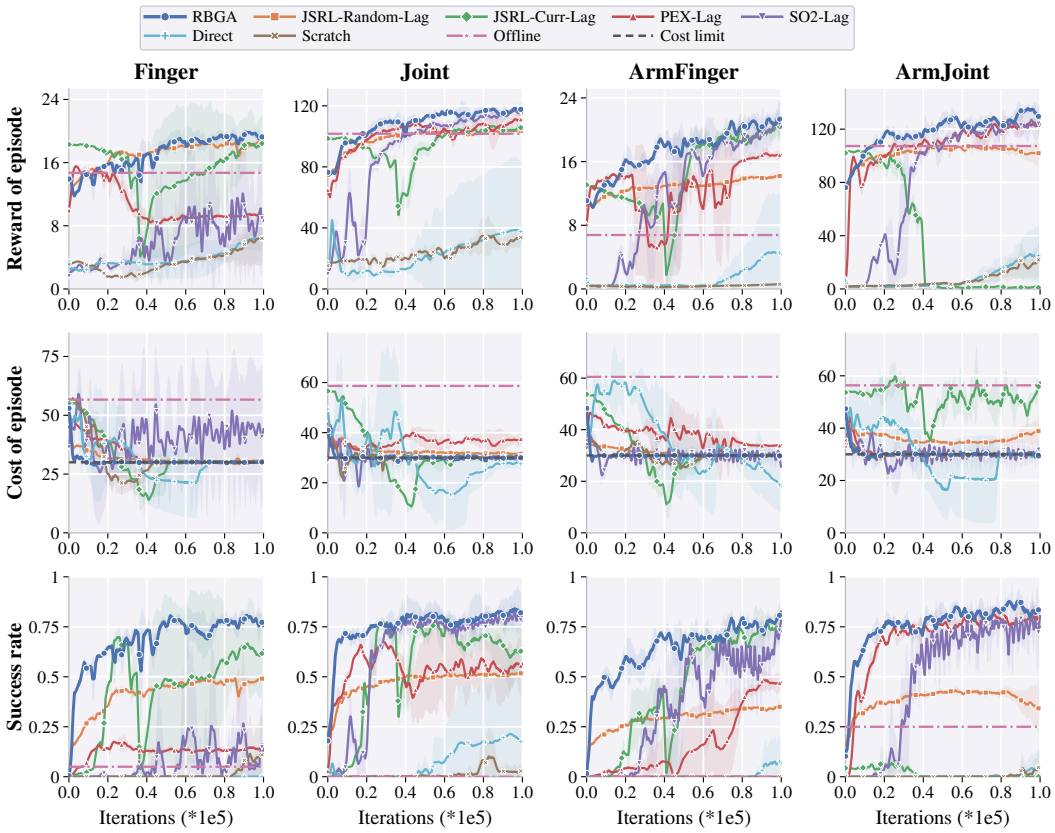


Figure 1: The figure depicts the reward, cost, and success rate curves of RBGA and the baseline algorithms in the comparison experiment. The shaded areas represent variance over three random seeds. The dashed line indicates the episode cost limit $d = 30$.

cost, and success rate curves of RBGA and the baseline algorithms on the four tasks. From the illustrated results, it can be observed that, compared with the baseline algorithms, RBGA maintains the episode cost close to the predefined cost limit while achieving higher rewards and success rates. Furthermore, RBGA demonstrates more stable performance compared with the other baseline algorithms during online training. Notably, RBGA achieves the best performance across the four tasks while maintaining favorable safety performance under the cost constraint. The above results indicate that RBGA consistently enables effective online policy improvement across various safe dexterous manipulation tasks while maintaining the episode cost around the predefined cost limit. We also evaluate RBGA on other manipulation tasks in Appendix B.3, and describe its deployment for real-world cube grasping in Appendix B.4.

5.3 Ablation Study

To evaluate the effect of different guide horizon scheduling strategies on online learning, we conduct ablation experiments across the four dexterous manipulation tasks. Specifically, we compare the proposed Linear and Step schedules with JSRL-Random and JSRL-Curriculum (Uchendu et al., 2023), as well as a Fixed schedule. For Linear, Step, and JSRL-Curriculum, guidance ends at 40k, 60k, or 80k online training iterations, whereas JSRL-Random and Fixed retain guide participation throughout training. Figure 2 illustrates the reward, cost, and success rate curves of the different scheduling strategies across the four tasks. From the illustrated results, it can be observed that the Linear and Step schedules generally provide stable policy improvement and achieve competitive reward and success rate. Notably, Linear-40k achieves consistently strong performance across the four tasks while maintaining favorable cost control. In contrast, several JSRL-Curriculum config-

urations exhibit large performance fluctuations or sharp degradation during online training, while JSRL-Random and Fixed also do not consistently match the performance of the decreasing-horizon schedules. These results indicate that sampling the guide horizon within a gradually decreasing maximum horizon provides more stable online learning than using a deterministic stage-wise guide horizon. This is because sampling the guide horizon within the current maximum horizon as it gradually decreases over training allows the online policy to take over at different episode steps, leading to smoother changes in the state distribution and reducing abrupt shifts between curriculum stages. We additionally evaluate the contribution of residual-budget arbitration in Appendix B.1.

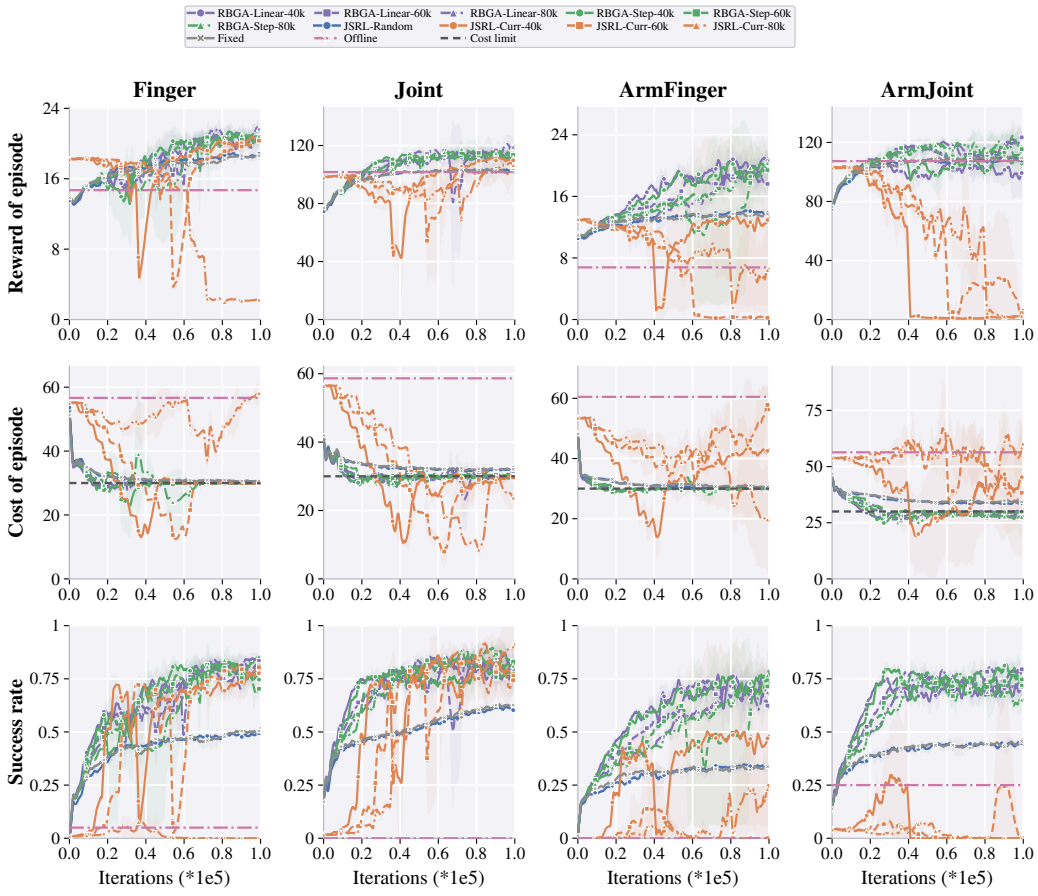


Figure 2: The figure depicts the reward, cost, and success rate curves under different guide horizon scheduling strategies. The shaded areas represent variance over three random seeds. The labels 40k, 60k, and 80k indicate the iterations at which guidance ends, while Random and Fixed retain guidance throughout training. The dashed line indicates the episode cost limit $d = 30$.

6 Conclusion

In this work, we study O2O safe RL for dexterous manipulation and propose Residual-Budget Guide Arbitration (RBGA). Specifically, we first use a decreasing guide horizon to control the participation of a fixed offline guide during online learning. Based on this, we introduce a residual budget to track the episode cost already incurred and select between the guide and online actions according to the remaining budget and their estimated cost-to-go. Furthermore, we provide theoretical analyses of the cost of the composite policy and the reward performance of the learned online policy. Finally, extensive comparison and ablation experiments on dexterous manipulation tasks demonstrate that RBGA achieves strong task performance while maintaining favorable safety performance.

AI Assistance Disclosure

In this work, we used generative AI tools to aid and polish writing, retrieve and discover relevant literature, support research ideation and execution, draft and revise portions of the manuscript, and assist in formulating and checking mathematical claims and proofs. The authors reviewed and verified all AI-assisted content, including literature references, technical descriptions, mathematical derivations, and proofs. We take responsibility for the final content of this work, including text, claims, results, and mathematical statements produced with the aid of generative AI.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 22–31. PMLR, 2017. URL https://proceedings.mlr.press/v70/achiam17a.html.

Eitan Altman. *Constrained Markov Decision Processes*. Chapman and Hall/CRC, Boca Raton, Florida, 1999.

Philip J. Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 1577–1594. PMLR, 2023. URL https://proceedings.mlr.press/v202/ball23a.html.

Chenyang Cao, Yucheng Xin, Silang Wu, Longxiang He, Zichen Yan, Junbo Tan, and Xueqian Wang. FOSP: Fine-tuning offline safe policy through world models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=dbuFJg7eaw.

Henry J. Charlesworth and Giovanni Montana. Solving challenging dexterous manipulation tasks with trajectory optimisation and reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1496–1506. PMLR, 2021. URL https://proceedings.mlr.press/v139/charlesworth21a.html.

Keru Chen, Honghao Wei, Zhigang Deng, and Sen Lin. Towards fast safe online reinforcement learning via policy finetuning. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=1SO7vmLFUq.

Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object re-orientation. In Aleksandra Faust, David Hsu, and Gerhard Neumann (eds.), *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 297–307. PMLR, 2022a. URL https://proceedings.mlr.press/v164/chen22a.html.

Yuanpei Chen, Tianhao Wu, Shengjie Wang, Xidong Feng, Jiechuan Jiang, Zongqing Lu, Stephen McAleer, Hao Dong, Song-Chun Zhu, and Yaodong Yang. Towards human-level bimanual dexterous manipulation with reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5150–5163, 2022b. doi: 10.52202/068431-0372. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/217a2a387f52c30755c37b0a73430291-Abstract-Datasets_and_Benchmarks.html.

Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A Lyapunov-based approach to safe reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/4fe5149039b52765bde64beb9f674940-Paper.pdf.

Gabriel Dulac-Arnold, Nir Levine, Daniel J. Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: Definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, 2021. doi: 10.1007/s10994-021-05961-4. URL https://link.springer.com/article/10.1007/s10994-021-05961-4.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062. PMLR, 2019. URL https://proceedings.mlr.press/v97/fujimoto19a.html.

Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(42):1437–1480, 2015. URL https://jmlr.org/papers/v16/garcia15a.html.

Yiran Geng, Jiaming Ji, Yuanpei Chen, Haoran Geng, Fangwei Zhong, and Yaodong Yang. ReDMan: Reliable dexterous manipulation with safe reinforcement learning. *Machine Learning*, 114 (8):189, 2025. doi: 10.1007/s10994-025-06825-x.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Binghao Huang, Yuanpei Chen, Tianyu Wang, Yuzhe Qin, Yaodong Yang, Nikolay Atanasov, and Xiaolong Wang. Dynamic handover: Throw and catch with bimanual hands. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1887–1902. PMLR, 2023. URL https://proceedings.mlr.press/v229/huang23d.html.

Abhineet Jain, Jack Kolb, and Harish Ravichandar. Constrained reinforcement learning for dexterous manipulation, 2023. URL https://arxiv.org/abs/2301.09766.

Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Juntao Dai, and Yaodong Yang. Safety-Gymnasium: A unified safe reinforcement learning benchmark. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2310.12567.

Jiaming Ji, Jiayi Zhou, Borong Zhang, Juntao Dai, Xuehai Pan, Ruiyang Sun, Weidong Huang, Yiran Geng, Mickel Liu, and Yaodong Yang. OmniSafe: An infrastructure for accelerating safe reinforcement learning research. *Journal of Machine Learning Research*, 25(285):1–6, 2024. URL https://jmlr.org/papers/v25/23-0681.html.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning*, pp. 267–274, 2002. URL https://homes.cs.washington.edu/~sham/papers/rl/aoarl.pdf.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-Learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy Q-Learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/c2073ffa77b5357a498057413bb09d3a-Paper.pdf.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-Learning for offline reinforcement learning. In *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.

Jongmin Lee, Cosmin Paduraru, Daniel J. Mankowitz, Nicolas Heess, Doina Precup, Kee-Eung Kim, and Arthur Guez. COptiDICE: Offline constrained reinforcement learning via stationary distribution correction estimation. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=FLA55mBee6Q.

Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic Q-Ensemble. In Aleksandra Faust, David Hsu, and Gerhard Neumann (eds.), *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1702–1712. PMLR, 2022b. URL https://proceedings.mlr.press/v164/lee22d.html.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020. URL https://arxiv.org/abs/2005.01643.

Jinning Li, Xinyi Liu, Banghua Zhu, Jiantao Jiao, Masayoshi Tomizuka, Chen Tang, and Wei Zhan. Guided online distillation: Promoting safe reinforcement learning by offline demonstration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7447–7454. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10611123.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1509.02971.

Zuxin Liu, Zijian Guo, Yihang Yao, Zhepeng Cen, Wenhao Yu, Tingnan Zhang, and Ding Zhao. Constrained decision transformer for offline safe reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 21611–21630. PMLR, 2023. URL https://proceedings.mlr.press/v202/liu23m.html.

Zuxin Liu, Zijian Guo, Haohong Lin, Yihang Yao, Jiacheng Zhu, Zhepeng Cen, Hanjiang Hu, Wenhao Yu, Tingnan Zhang, Jie Tan, and Ding Zhao. Datasets and benchmarks for offline safe reinforcement learning. *Journal of Data-centric Machine Learning Research*, 2024. URL https://data.mlr.press/assets/pdf/v01-12.pdf.

Qin-Wen Luo, Ming-Kun Xie, Ye-Wen Wang, and Sheng-Jun Huang. Optimistic critic reconstruction and constrained fine-tuning for general offline-to-online RL. In *Advances in Neural Information Processing Systems*, volume 37, pp. 108167–108207, 2024. doi: 10.52202/079017-3435. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/c3b3c5f9297ad0012cfe7188c34cea0e-Abstract-Conference.html.

Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1101–1112. PMLR, 2020. URL https://proceedings.mlr.press/v100/nagabandi20a.html.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. AWAC: Accelerating online reinforcement learning with offline datasets, 2020. URL https://arxiv.org/abs/2006.09359.

Mitsuhiko Nakamoto, Simon Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-QL: Calibrated offline RL pretraining for efficient online fine-tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 62244–62269. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/c44a04289beaf0a7d968a94066a1d696-Abstract-Conference.html.

OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Józefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020. doi: 10.1177/0278364919887447. URL https://journals.sagepub.com/doi/10.1177/0278364919887447.

Haozhi Qi, Ashish Kumar, Roberto Calandra, Yi Ma, and Jitendra Malik. In-hand object rotation via rapid motor adaptation. In *Proceedings of the 6th Conference on Robot Learning*, volume 205

of *Proceedings of Machine Learning Research*, pp. 1722–1732. PMLR, 2023a. URL https://proceedings.mlr.press/v205/qi23a.html.

Haozhi Qi, Brent Yi, Sudharshan Suresh, Mike Lambeta, Yi Ma, Roberto Calandra, and Jitendra Malik. General in-hand object rotation with vision and touch. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2549–2564. PMLR, 2023b. URL https://proceedings.mlr.press/v229/qi23a.html.

Yuzhe Qin, Binghao Huang, Zhao-Heng Yin, Hao Su, and Xiaolong Wang. DexPoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 594–605. PMLR, 2023. URL https://proceedings.mlr.press/v205/qin23a.html.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Proceedings of Robotics: Science and Systems*, 2018. doi: 10.15607/RSS.2018.XIV.049.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. Technical report, OpenAI, 2019. URL https://cdn.openai.com/safexp-short.pdf.

Harsh Satija, Philip Amortila, and Joelle Pineau. Constrained Markov decision processes via backward value functions. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8502–8511. PMLR, 2020. URL https://proceedings.mlr.press/v119/satija20a.html.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by PID lagrangian methods. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9133–9143. PMLR, 2020. URL https://proceedings.mlr.press/v119/stooke20a.html.

Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. Reward constrained policy optimization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkfrvsA9FX.

Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E. Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery RL: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6(3): 4915–4922, 2021. doi: 10.1109/LRA.2021.3070252. URL https://ieeexplore.ieee.org/document/9392290/.

Ikechukwu Uchendu, Ted Xiao, Yao Lu, Banghua Zhu, Mengyuan Yan, Joséphine Simon, Matthew Bennice, Chuyuan Fu, Cong Ma, Jiantao Jiao, Sergey Levine, and Karol Hausman. Jump-start reinforcement learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 34556–34583. PMLR, 2023. URL https://proceedings.mlr.press/v202/uchendu23a.html.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. FoundationPose: Unified 6D pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17868–17879, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wen_FoundationPose_Unified_6D_Pose_Estimation_and_Tracking_of_Novel_Objects_CVPR_2024_paper.html.

Haoran Xu, Xianyuan Zhan, and Xiangyu Zhu. Constraints penalized Q-learning for safe offline reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8): 8753–8760, 2022. doi: 10.1609/aaai.v36i8.20855. URL https://ojs.aaai.org/index.php/AAAI/article/view/20855.

Zishun Yu and Xinhua Zhang. Actor-critic alignment for offline-to-online reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 40452–40474. PMLR, 2023. URL https://proceedings.mlr.press/v202/yu23k.html.

Haoqi Yuan, Bohan Zhou, Yuhui Fu, and Zongqing Lu. Cross-embodiment dexterous grasping with reinforcement learning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=twIPSx9qHn.

Weishu Zhan and Peter Chin. Safe multi-agent reinforcement learning for bimanual dexterous manipulation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12420–12427. IEEE, 2024. doi: 10.1109/IROS58592.2024.10801490. URL https://ieeexplore.ieee.org/document/10801490.

Haichao Zhang, Wei Xu, and Haonan Yu. Policy expansion for bridging offline-to-online reinforcement learning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=-Y34L45JR6z.

Yinmin Zhang, Jie Liu, Chuming Li, Yazhe Niu, Yaodong Yang, Yu Liu, and Wanli Ouyang. A perspective of Q-Value estimation on offline-to-online reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(15):16908–16916, 2024. doi: 10.1609/aaai.v38i15.29633. URL https://ojs.aaai.org/index.php/AAAI/article/view/29633.

Henry Zhu, Abhishek Gupta, Aravind Rajeswaran, Sergey Levine, and Vikash Kumar. Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3651–3657. IEEE, 2019. doi: 10.1109/ICRA.2019.8794102.

A Additional Experimental Details

A.1 Task Descriptions

To evaluate the performance of RBGA in various safe dexterous manipulation tasks with different manipulation settings and joint-range constraints, we select four tasks including *Finger*, *Joint*, *ArmFinger*, and *ArmJoint* from ReDMan (Geng et al., 2025) as the experimental tasks in this work.

Finger. Two fixed-base dexterous hands cooperate to transfer a ball to a target position. The safety constraint restricts the ranges of motion of three index-finger joints.

Joint. The manipulation objective and hand configuration are the same as in Finger, but the safety constraint applies to a single index-finger joint.

ArmFinger. The object is thrown from a vertically oriented hand and caught by a second hand with its palm facing upward. Multiple index-finger joints are subject to range-of-motion constraints.

ArmJoint. This task uses the same throwing-and-catching setting as ArmFinger, with the safety constraint imposed on a single index-finger joint.

A.2 Offline Guide and Online Learning

Unless otherwise stated, RBGA uses implicit Q-learning (IQL) (Kostrikov et al., 2022) to train the offline guide and SAC-Lagrangian (SAC-Lag) (Haarnoja et al., 2018; Ji et al., 2024) for online learning. We train the offline policy π_g on a fixed dataset $\mathcal{D}_{\text{off}}$ containing 1,000,000 transitions per task and keep it fixed during online learning. The critics and the auxiliary cost estimator associated with the online policy π_o are initialized independently of the guide policy.

The composite policy π collects transitions into the online replay buffer $\mathcal{B}$. The guide policy remains fixed throughout online learning, whereas the online policy is continually updated using transitions collected in the online replay buffer. Each collection comprises eight steps in each of 512 parallel environments. Learning begins after the buffer contains 500,000 transitions; each subsequent collection is followed by ten training iterations, up to the total of N. Each iteration updates the SAC-Lag learner and then the auxiliary estimator using a fresh minibatch. Table 1 summarizes the architecture and training settings.

A.3 Online SAC-Lag Updates

Critic updates. For a transition sampled from $\mathcal{B}$, the reward and cost TD targets are

$$\begin{aligned} y_R &= r + \gamma(1-\delta) \min_{j\in\{1,2\}} Q_{\bar{\phi}_{R,j}}(s', a'), \\ y_C &= c + \gamma(1-\delta) \max_{j\in\{1,2\}} Q_{\bar{\phi}_{C,j}}(s', a'). \end{aligned} \tag{15}$$

where (s, a, r, c, s', δ) is a sampled transition, $a' \sim \pi_o(\cdot \mid s')$, and δ indicates episode termination. The functions $Q_{\phi_{R,j}}$ and $Q_{\phi_{C,j}}$, $j \in \{1, 2\}$, denote the reward and cost critics, respectively, and an overbar denotes the corresponding target-network parameters.

Policy and temperature updates. The online policy minimizes

$$\mathcal{L}_{\pi_o} = \frac{1}{1+\lambda} \mathbb{E}_{\substack{s\sim\mathcal{B} \\ a\sim\pi_o(\cdot|s)}} \left[\alpha \log \pi_o(a \mid s) - \min_{j\in\{1,2\}} Q_{\phi_{R,j}}(s,a) + \lambda \max_{j\in\{1,2\}} Q_{\phi_{C,j}}(s,a) \right]. \tag{16}$$

where s is sampled from replay, a is a newly sampled online action, and $\lambda \geq 0$ is the Lagrange multiplier. Critic parameters, α, and λ are held fixed during this update. We adjust α by optimizing $\log \alpha$ with a target entropy equal to the negative action dimension.

Multiplier and target-network updates. Once learning has started, each collection with completed episodes is followed by one Adam step minimizing

$$\mathcal{L}_\lambda = -\lambda\big(\bar{C}_{\text{ep}} - d\big), \tag{17}$$

where $\bar{C}_{\text{ep}}$ is the mean cumulative cost of the episodes completed in that collection. We clip λ to $[0, 5]$ after this update. Each training iteration updates the reward critics, cost critics, online policy, and entropy temperature, then soft-updates the target parameters with weight 0.005 on the current parameters and 0.995 on their previous target values.

Table 1: RBGA hyperparameters used in the main experiments. The online SAC-Lag cost critic and the auxiliary cost estimator are separate networks.

Parameter	Value
Network architecture	
Hidden layers of each actor and critic	$[1024, 1024, 512]$
Hidden activation	ELU
Online policy	Tanh-squashed Gaussian
Reward critics / SAC-Lag cost critics	2 / 2
Auxiliary cost critics	2
Online optimization	
Optimizer	Adam
Actor / reward critic / cost critic learning rate	$1e-3$
Entropy temperature learning rate	$1e-3$
Initial entropy temperature	0.01
Target entropy	Negative action dimension
SAC-Lag TD discount	0.99
Target soft update coefficient	0.005
Maximum gradient norm	5
Initial Lagrange multiplier	0.001
Lagrange multiplier learning rate	$1e-3$
Lagrange multiplier bounds	$[0, 5]$
Data collection and replay	
Parallel environments	512
Steps per environment per collection	8
Training iterations per collection	10
Total training iterations N	100,000
Replay buffer capacity	1,000,000 transitions
Minimum replay size before learning	500,000 transitions
Batch size	4096
Replay sampling	Uniform
Guide scheduling and arbitration	
Episode horizon H / initial guide horizon H_0	75 / 75
Guide horizon schedule	Linear
Guidance cutoff N_g	40,000 iterations
Episode cost limit d	30
Comparison margin ζ	1
Residual-budget / auxiliary TD discount γ	1.0
Auxiliary estimator warm-up	2000 updates
Auxiliary estimator training	
Learning rate	$1e-3$
Minibatch size	4096
Updates per online training iteration	1
Sample retention probability	0.8
Regression loss	Smooth L1 (threshold 1)
Arbitration estimate	Non-negative target mean
Target soft update coefficient	0.005

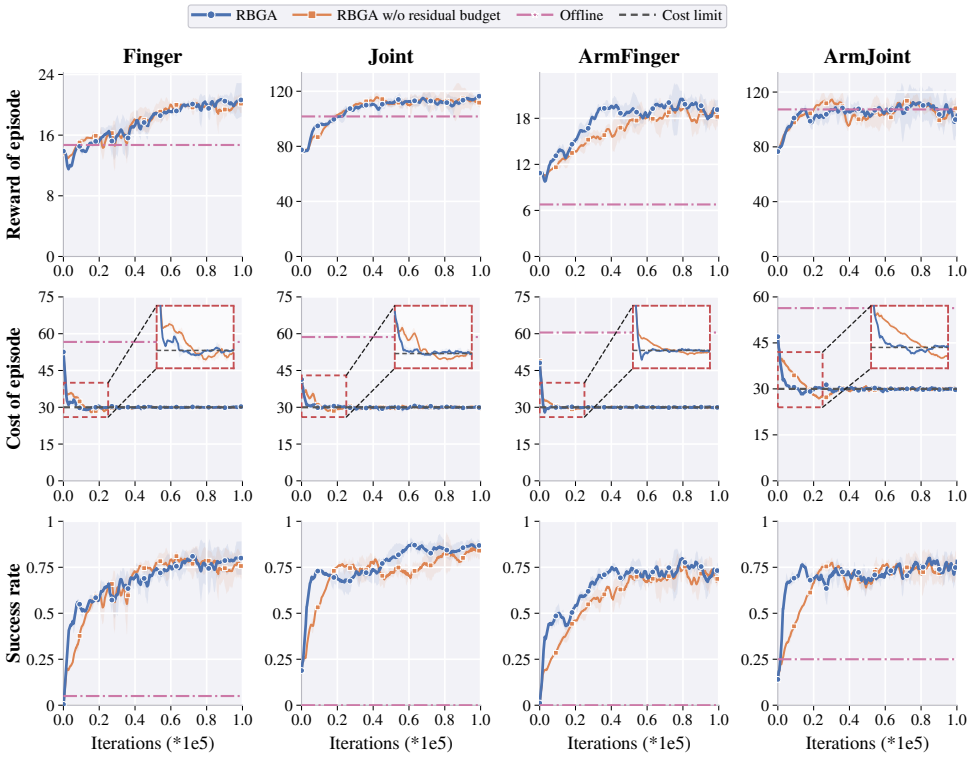


Figure 3: The figure depicts the reward, cost, and success rate curves of RBGA and RBGA without residual-budget arbitration on four tasks. The shaded areas on the curves represent the variance obtained from online testing conducted with three random seeds. The dashed line indicates the episode cost limit $d = 30$.

A.4 Auxiliary Cost-to-Go Estimator

Candidate evaluation. We train two auxiliary cost estimators to estimate the cost-to-go in Eq. (5). Both take the state and action as input. Arbitration uses the non-negative mean of their target predictions:

$$\widehat{Q}_C^{\pi_o}(s,a) = \left[\frac{1}{2}\sum_{j=1}^{2} Q_{\bar{\psi}_j}(s,a)\right]_+, \tag{18}$$

where $Q_{\bar{\psi}_j}$ denotes the target network of the j-th auxiliary cost critic, and $[\cdot]_+$ clips negative predictions to zero.

Estimator update. The auxiliary cost estimators are trained using the TD target

$$y^{\text{aux}} = c + \gamma(1-\delta)\widehat{Q}_C^{\pi_o}(s',a'), \qquad a' \sim \pi_o(\cdot \mid s'), \tag{19}$$

where (s, a, c, s', δ) is a transition sampled from $\mathcal{B}$, and γ is the discount factor used in Eq. (5). The target estimates the cost of executing the stored action and following the current online policy thereafter. The auxiliary estimators are optimized using the Huber loss, with the remaining training details provided in Table 1.

Warm-up. The auxiliary estimator is trained during the first 2,000 online training iterations, while actions are chosen using the guide horizon without budget arbitration. Once both the online learner and auxiliary estimator have completed 2,000 updates, action selection follows Eq. (6).

A.5 Guide Horizon Schedules

The linear schedule is defined in Eq. (3) in Section 4. For the step schedule, we predefine decreasing horizon values from H_0 to zero and assign them to approximately evenly spaced training stages over the first N_g iterations. The value of H_i is constant within each stage. By default, the horizon values are obtained by evenly spacing values between H_0 and zero. Both schedules control guide participation without changing the task horizon H.

B Additional Experimental Results

B.1 Effect of Residual-Budget Arbitration

To evaluate the contribution of residual-budget arbitration, we disable the residual-budget mechanism while keeping the remaining settings unchanged. As shown in Figure 3, RBGA satisfies the cost constraint earlier without degrading reward. These results indicate that residual-budget arbitration achieves a more favorable safety–performance trade-off during the early stage of online training. This suggests that incorporating the remaining episode budget helps RBGA select actions with lower predicted cost-to-go while preserving task performance.

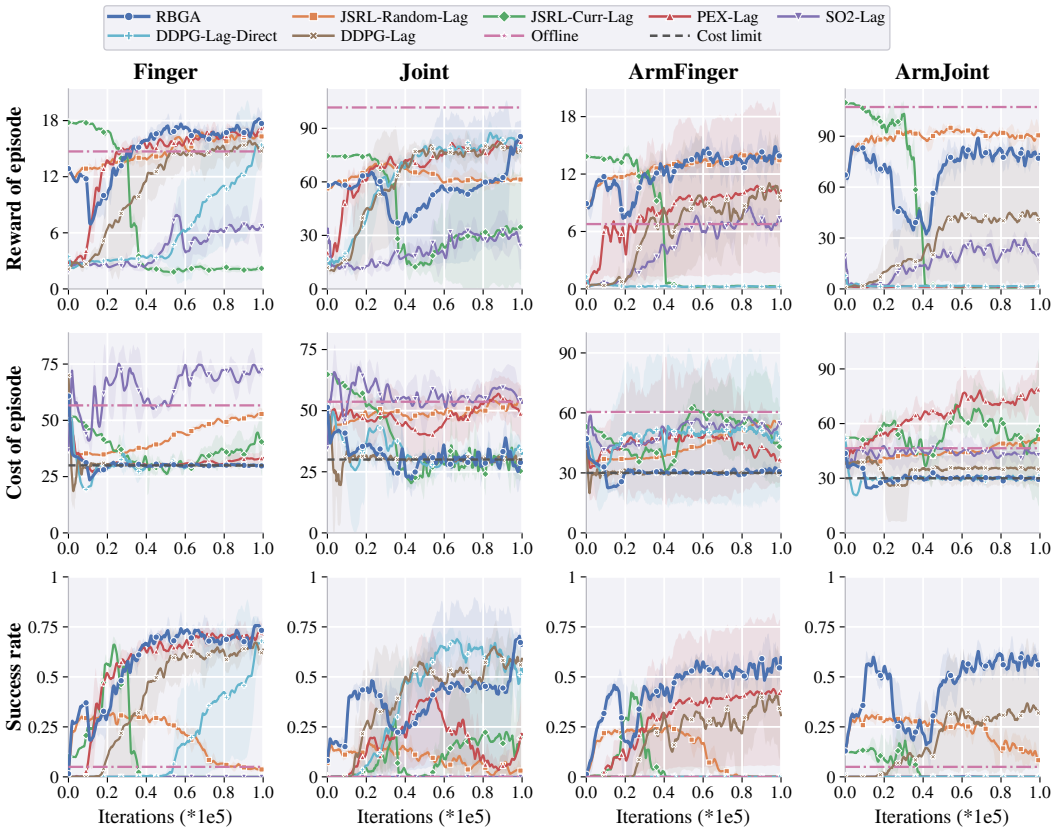


Figure 4: The figure depicts the reward, cost, and success rate curves of RBGA and the baseline algorithms using DDPG-Lag as the online learner. The shaded areas on the curves represent the variance obtained from online testing conducted with three random seeds. The dashed line indicates the episode cost limit $d = 30$.

B.2 Results with a DDPG-Lag Online Learner

To evaluate the performance of RBGA with a different online RL algorithm, we conduct comparative experiments using DDPG-Lag as the online learner. Specifically, we replace SAC-Lag with DDPG-Lag for RBGA and all corresponding baselines while retaining their original offline-to-online mechanisms. The remaining data collection, guide scheduling, and arbitration settings are kept unchanged.

Figure 4 illustrates the reward, cost, and success rate curves of RBGA and the baseline algorithms on the four tasks. From the illustrated results, it can be observed that RBGA maintains the episode cost close to the predefined cost limit while achieving competitive reward and success rate across the four tasks. Furthermore, RBGA demonstrates stable policy improvement during online training, whereas several baseline algorithms exhibit higher costs or lower task performance. Although the reward of RBGA temporarily decreases during the early stage of training on *ArmJoint*, it subsequently recovers while maintaining a relatively low episode cost and a high success rate. The above results indicate that RBGA can be effectively combined with different constrained online RL algorithms.

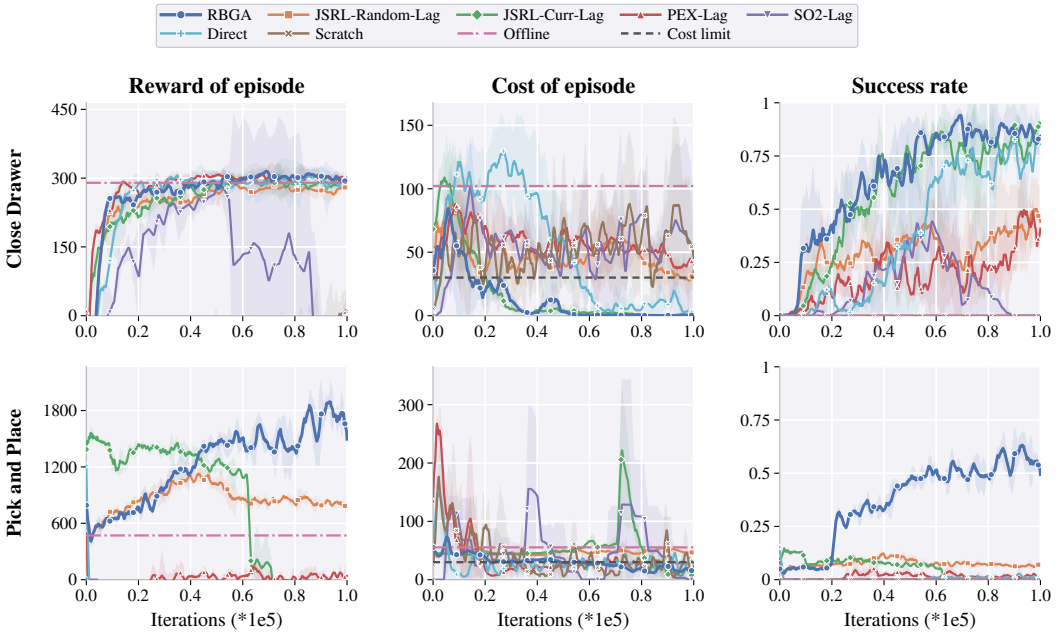


Figure 5: The figure depicts the reward, cost, and success rate curves of RBGA and the baseline algorithms on the *Close Drawer* and *Pick and Place* tasks. The shaded areas on the curves represent the variance obtained from online testing conducted with three random seeds. The dashed line indicates the episode cost limit $d = 30$.

B.3 Results on Additional Manipulation Tasks

We further evaluate RBGA on two additional manipulation tasks including *Close Drawer* and *Pick and Place*, two Safety-Gymnasium manipulation tasks from Safe Policy Optimization (Ji et al., 2023). Both use a mobile base equipped with a Franka arm and a gripper. *Close Drawer* requires the robot to close a drawer while keeping the base outside a restricted region near the cabinet, whereas *Pick and Place* requires the robot to move an object from a lower platform to a higher location while avoiding the restricted region between the platforms. RBGA and all six baselines use SAC-Lag for online learning.

Figure 5 illustrates the reward, cost, and success rate curves of RBGA and the baseline algorithms on the two tasks. From the illustrated results, it can be observed that RBGA consistently improves task performance during online training while maintaining the episode cost within or close to the

predefined cost limit. On *Close Drawer*, RBGA achieves high reward and success rate while reducing the episode cost toward the cost limit, whereas several baseline algorithms exhibit higher costs or lower success rates during training. On *Pick and Place*, RBGA continues to improve beyond the offline policy and achieves higher reward and success rate than the compared methods, while several baselines show limited improvement or performance degradation. The above results indicate that RBGA can effectively leverage offline guidance for safe online policy improvement across different manipulation tasks.

B.4 Real-world Deployment

To further evaluate the practical applicability of RBGA, we deploy the policy trained in simulation environment directly on a real-world robotic platform consisting of a RealMan RM65 robotic arm and a Linker Hand L20 dexterous hand. We consider a *Cube Grasping* task, where the same arm–hand platform is modeled in simulation for policy training. For real-world execution, an external Intel RealSense D435 RGB-D camera captures the scene, and FoundationPose (Wen et al., 2024) estimates the 6D pose of the target object. The estimated object pose is transformed into the robot reference frame and combined with the robot proprioception to construct the same policy observation used during IsaacGym training. The resulting observation is directly fed into the learned policy to generate control commands for the robotic arm and dexterous hand.

As shown in Figure 6, we present the corresponding grasping behaviors in simulation and on the physical robot. The policy learned with RBGA can be transferred to the physical platform and successfully execute the cube grasping behavior while maintaining stable and safety-aware manipulation. This result demonstrates the potential of RBGA for transferring safe dexterous manipulation policies from simulation to real robotic systems.

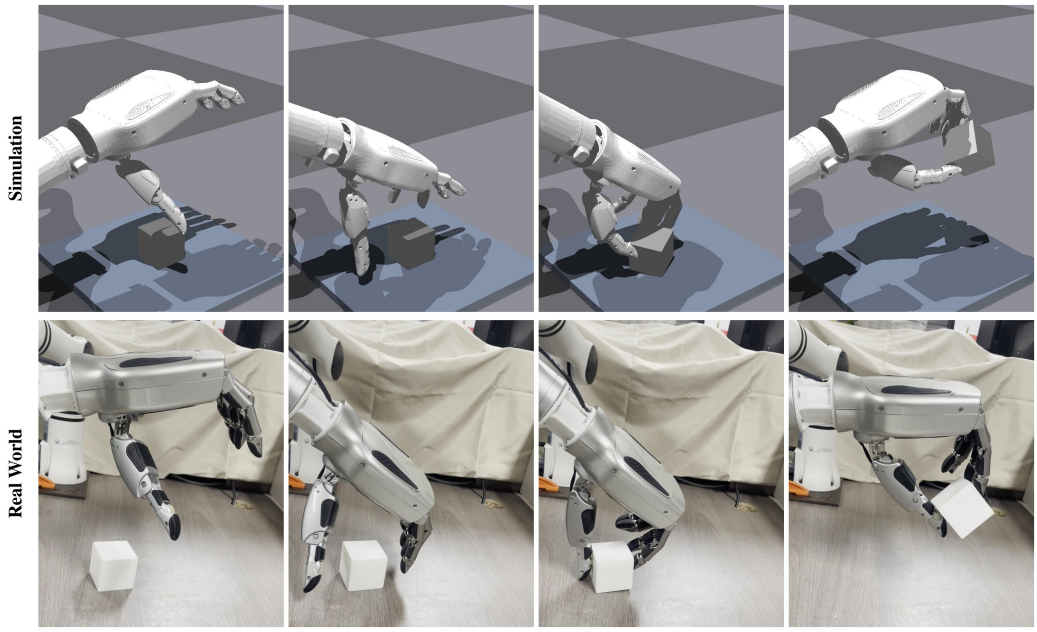


Figure 6: Simulation and real-world deployment of RBGA on the *Cube Grasping* task. Top: simulation. Bottom: the physical robot. The platform consists of a RealMan RM65 robotic arm equipped with a Linker Hand L20 dexterous hand.

C Proofs and Discussions

We provide the proofs of the cost and reward bounds in Section 4.2. We first derive the Bellman recursion used in Section 4.1, and then prove the local and episode cost bounds. Subsequently, we discuss the state coverage and policy advantage conditions and prove the reward performance bound.

C.1 Proof of the Cost Bellman Recursion

For a fixed online policy π_o, the cost state values at $t = 0, \dots, H-1$ and the terminal value are given as:

$$V_C^{\pi_o}(s_t) = \mathbb{E}_{a_t^o \sim \pi_o(\cdot|s_t)}[Q_C^{\pi_o}(s_t, a_t^o)], \qquad V_C^{\pi_o}(s_H) = 0, \tag{20}$$

where $V_C^{\pi_o}(s_t)$ denotes the expected discounted cost-to-go from state s_t at time t under π_o. The cost-to-go in Eq. (5) satisfies the following Bellman recursion:

$$Q_C^{\pi_o}(s_t, a_t) = c(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim P(\cdot|s_t, a_t)}[V_C^{\pi_o}(s_{t+1})]. \tag{21}$$

Proof. By separating the immediate cost from the remaining discounted cost in Eq. (5), we obtain:

$$\begin{aligned} Q_C^{\pi_o}(s_t, a_t) &= c(s_t, a_t) + \gamma \mathbb{E}_{\pi_o}\left[\sum_{j=t+1}^{H-1} \gamma^{j-(t+1)} c_j \,\middle|\, s_t, a_t\right] \\ &= c(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim P(\cdot|s_t, a_t)}\left[V_C^{\pi_o}(s_{t+1})\right], \end{aligned} \tag{22}$$

The first equality uses $c_t = c(s_t, a_t)$, and the second equality follows by conditioning on the next state s_{t+1} and using Eq. (20). At $t = H-1$, the remaining sum is empty and $V_C^{\pi_o}(s_H) = 0$, so the same equality holds. Therefore, the cost-to-go satisfies Eq. (21). □

C.2 Proof and Discussion of Proposition 4.2

Assumption 4.1. For a fixed online policy π_o and cost estimator $\widehat{Q}_C^{\pi_o}$, there exists a finite constant $\varepsilon_C \geq 0$ such that, at each time step $t < \min\{h, H_i\}$, the cost estimates of both candidate actions in Eq. (6) satisfy:

$$\left|\widehat{Q}_C^{\pi_o}(s_t, a) - Q_C^{\pi_o}(s_t, a)\right| \leq \varepsilon_C, \qquad \forall a \in \{a_t^g, a_t^o\}, \tag{23}$$

where a ranges over the guide and online candidates, and ε_C is a uniform bound on their estimation errors.

Proposition 4.2. Under Assumption 4.1, consider a time step $t < \min\{h, H_i\}$. For a selected action satisfying $\gamma^t \widehat{Q}_C^{\pi_o}(s_t, a_t) \leq b_t$, the accumulated discounted cost and expected continuation cost satisfy:

$$\sum_{j=0}^{t-1} \gamma^j c_j + \gamma^t Q_C^{\pi_o}(s_t, a_t) \leq d + \gamma^t \varepsilon_C \leq d + \varepsilon_C. \tag{24}$$

For every guide action selected by Eq. (6), the difference in cost-to-go satisfies:

$$Q_C^{\pi_o}(s_t, a_t^g) - Q_C^{\pi_o}(s_t, a_t^o) \leq \zeta + 2\varepsilon_C. \tag{25}$$

Proof. By applying the estimation error bound to the selected action and substituting the budget condition, we obtain:

$$\begin{aligned} \sum_{j=0}^{t-1} \gamma^j c_j + \gamma^t Q_C^{\pi_o}(s_t, a_t) &\leq \sum_{j=0}^{t-1} \gamma^j c_j + \gamma^t \widehat{Q}_C^{\pi_o}(s_t, a_t) + \gamma^t \varepsilon_C \\ &\leq \sum_{j=0}^{t-1} \gamma^j c_j + b_t + \gamma^t \varepsilon_C \\ &= d + \gamma^t \varepsilon_C \leq d + \varepsilon_C, \end{aligned} \tag{26}$$

where the first inequality follows from $Q_C^{\pi_o}(s_t, a_t) \leq \widehat{Q}_C^{\pi_o}(s_t, a_t) + \varepsilon_C$ in Assumption 4.1, the second uses $\gamma^t \widehat{Q}_C^{\pi_o}(s_t, a_t) \leq b_t$, the equality follows from Eq. (4), and the final inequality uses $\gamma^t \leq 1$. Therefore, Eq. (8) holds.

To derive the guide comparison bound, we consider the branches of Eq. (6) that select a guide action. If the selected guide action passes the budget condition, the second branch of the selection rule gives:

$$\widehat{q}_t^g - \widehat{q}_t^o \leq \min\{\gamma^t \zeta, b_t - \widehat{q}_t^o\}. \tag{27}$$

This inequality covers the cases in which both candidates pass the budget condition and in which only the guide candidate passes. If the selected guide action fails the budget condition, the online candidate also fails, and the minimum-cost fallback gives $\widehat{q}_t^g - \widehat{q}_t^o \leq 0$. Since $\gamma^t \zeta \geq 0$, the estimate difference in both cases satisfies:

$$\widehat{q}_t^g - \widehat{q}_t^o \leq \min\{\gamma^t \zeta, [b_t - \widehat{q}_t^o]_+\}. \tag{28}$$

where $[b_t - \widehat{q}_t^o]_+$ is the non-negative budget remaining after accounting for the online candidate's estimated discounted cost.

By Assumption 4.1, the guide candidate satisfies $Q_C^{\pi_o}(s_t, a_t^g) \leq \widehat{Q}_C^{\pi_o}(s_t, a_t^g) + \varepsilon_C$, while the online candidate satisfies $Q_C^{\pi_o}(s_t, a_t^o) \geq \widehat{Q}_C^{\pi_o}(s_t, a_t^o) - \varepsilon_C$. Applying these two bounds to the cost-to-go difference gives:

$$\begin{aligned}
&\gamma^t \left[Q_C^{\pi_o}(s_t, a_t^g) - Q_C^{\pi_o}(s_t, a_t^o)\right] \\
&\quad \leq \gamma^t \left[\widehat{Q}_C^{\pi_o}(s_t, a_t^g) + \varepsilon_C - \left(\widehat{Q}_C^{\pi_o}(s_t, a_t^o) - \varepsilon_C\right)\right] \\
&\quad = \gamma^t \left[\widehat{Q}_C^{\pi_o}(s_t, a_t^g) - \widehat{Q}_C^{\pi_o}(s_t, a_t^o)\right] + 2\gamma^t \varepsilon_C \\
&\quad = \widehat{q}_t^g - \widehat{q}_t^o + 2\gamma^t \varepsilon_C \\
&\quad \leq 2\gamma^t \varepsilon_C + \min\{\gamma^t \zeta, [b_t - \widehat{q}_t^o]_+\} \\
&\quad \leq \gamma^t (\zeta + 2\varepsilon_C).
\end{aligned} \tag{29}$$

The second equality uses the definitions of $\widehat{q}_t^g$ and $\widehat{q}_t^o$. The second inequality follows from the preceding bound on their difference, and the final inequality uses $\min\{\gamma^t \zeta, [b_t - \widehat{q}_t^o]_+\} \leq \gamma^t \zeta$. Dividing both sides by $\gamma^t > 0$ yields Eq. (9).

□

C.3 Proof and Discussion of Proposition 4.3

Proposition 4.3. Under Assumption 4.1, fix π_o, $\widehat{Q}_C^{\pi_o}$, and H_i throughout an episode, and sample $h \sim \mathrm{Unif}\{0, \ldots, H_i\}$ independently of the initial state. The composite policy π satisfies:

$$[J_C(\pi) - d]_+ \leq [J_C(\pi_o) - d]_+ + (\zeta + 2\varepsilon_C) \sum_{t=0}^{H_i - 1} \gamma^t \frac{H_i - t}{H_i + 1}. \tag{30}$$

Proof. The composite policy π may depend on the interaction history through b_t and on the sampled horizon h, while the online candidate is sampled from $\pi_o(\cdot \mid s_t)$.

To compare the expected discounted costs of π and π_o, we use the finite-horizon cost form of the performance difference lemma (Kakade & Langford, 2002). With both policies starting from $s_0 \sim \rho_0$, we obtain:

$$J_C(\pi) - J_C(\pi_o) = \sum_{t=0}^{H-1} \gamma^t \mathbb{E}_\pi \left[Q_C^{\pi_o}(s_t, a_t) - V_C^{\pi_o}(s_t)\right], \tag{31}$$

where the expectation is over trajectories generated by π, including the sampled guide horizon and candidate actions. Taking expectations in Eq. (21) and summing the discounted differences cancels the intermediate value terms. Using $V_C^{\pi_o}(s_H) = 0$ and $\mathbb{E}_{s_0 \sim \rho_0}[V_C^{\pi_o}(s_0)] = J_C(\pi_o)$ then gives Eq. (31).

For RBGA, the online candidate is sampled as $a_t^o \sim \pi_o(\cdot \mid s_t)$. Conditional on the history before candidate sampling, its expected cost-to-go equals $V_C^{\pi_o}(s_t)$ by Eq. (20). Substituting this relation into Eq. (31) gives:

$$J_C(\pi) - J_C(\pi_o) = \sum_{t=0}^{H-1} \gamma^t \mathbb{E}_\pi \left[Q_C^{\pi_o}(s_t, a_t) - Q_C^{\pi_o}(s_t, a_t^o) \right]. \tag{32}$$

When the online candidate is selected, the action-value difference is zero. When the guide candidate is selected, Eq. (9) bounds this difference by $\zeta + 2\varepsilon_C$. By substituting these bounds into Eq. (32), we obtain:

$$J_C(\pi) - J_C(\pi_o) \le (\zeta + 2\varepsilon_C) \sum_{t=0}^{H-1} \gamma^t \Pr_\pi(I_t^g = 1), \tag{33}$$

where $I_t^g \in \{0, 1\}$ equals one when the guide candidate is selected under Eq. (6) before episode termination, including selection by the fallback, and zero otherwise. The probability $\Pr_\pi(I_t^g = 1)$ includes the randomness of the trajectory, candidate actions, and sampled horizon. Since guide selection requires $t < h$ and h is uniform on $\{0, \dots, H_i\}$, the discounted sum of these probabilities satisfies:

$$\sum_{t=0}^{H-1} \gamma^t \Pr_\pi(I_t^g = 1) \le \sum_{t=0}^{H-1} \gamma^t \Pr(h > t) = \sum_{t=0}^{H_i-1} \gamma^t \frac{H_i - t}{H_i + 1}. \tag{34}$$

Combining this inequality with Eq. (33), taking positive parts, and using $[u + v]_+ \le [u]_+ + v$ for $v \ge 0$ completes the proof:

$$\begin{aligned} [J_C(\pi) - d]_+ &\le \left[J_C(\pi_o) - d + (\zeta + 2\varepsilon_C) \sum_{t=0}^{H_i-1} \gamma^t \frac{H_i - t}{H_i + 1} \right]_+ \\ &\le [J_C(\pi_o) - d]_+ + (\zeta + 2\varepsilon_C) \sum_{t=0}^{H_i-1} \gamma^t \frac{H_i - t}{H_i + 1}. \end{aligned} \tag{35}$$

□

Proposition 4.3 establishes the episode cost bound with the online policy, cost estimator, and maximum guide horizon fixed throughout the episode. In online learning, an episode may instead span multiple policy updates. These updates can change the expected remaining cost, so their effect must be included when bounding the cost of the resulting interaction. Proposition C.1 extends the analysis to this setting by adding a term for increases in the online policy's cost value. Under the fixed-policy conditions of Proposition 4.3, this term vanishes and the original bound is recovered.

Proposition C.1 (Cost bound with policy updates)**.** *Let i_t be the current training iteration before action selection at episode step t, and let $\pi_o^{(t)}$ denote the online policy available at that step. The superscript (t) distinguishes policy versions only in this extension. Fix the initial learner state, including i_0 and $\pi_o^{(0)}$, before sampling $s_0 \sim \rho_0$ and $h \sim \mathrm{Unif}\{0, \dots, H_{i_0}\}$ independently. Keep h fixed throughout the episode. At each step, the current online policy and cost estimator are determined before candidate sampling. Assume Eq. (7) holds with the same ε_C for both candidates at every step satisfying $t < \min\{h, H_{i_t}\}$. If Eq. (6) is used throughout the episode with the current policy, estimator, and maximum guide horizon, then:*

$$\begin{aligned} \mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t c_t \right] \le\ & J_C(\pi_o^{(0)}) \\ &+ \mathbb{E}\left[\sum_{t=0}^{H-1} I_t^g \left(2\gamma^t \varepsilon_C + \min\{\gamma^t \zeta, [b_t - \widehat{q}_t^o]_+\} \right) \right] \\ &+ \sum_{t=1}^{H-1} \gamma^t \mathbb{E}\left[\left[V_C^{\pi_o^{(t)}}(s_t) - V_C^{\pi_o^{(t-1)}}(s_t) \right]_+ \right], \end{aligned} \tag{36}$$

where $\widehat{q}_t^o = \gamma^t \widehat{Q}_C^{\pi_o^{(t)}}(s_t, a_t^o)$ uses the current cost estimator, and I_t^g indicates guide selection. The expectations include randomness in interaction and policy updates, conditional on the initial learner state. Both value functions in the last term are evaluated at the same state s_t and episode step t. For episodes that terminate before H, remaining costs and terminal-state values are set to zero.

Proof. Applying the Bellman recursion in Eq. (21) to the current policy, held fixed for each value evaluation, and taking expectations gives:

$$\begin{aligned} &\mathbb{E}\left[Q_C^{\pi_o^{(t)}}(s_t, a_t) - V_C^{\pi_o^{(t)}}(s_t)\right] \\ &\quad = \mathbb{E}\left[c_t + \gamma V_C^{\pi_o^{(t)}}(s_{t+1}) - V_C^{\pi_o^{(t)}}(s_t)\right]. \end{aligned} \tag{37}$$

Multiplying by γ^t and summing over t leaves the value differences between successive policies at the intermediate states. Moreover, conditional on the history before candidate sampling, $a_t^o \sim \pi_o^{(t)}(\cdot \mid s_t)$ implies that its expected action value equals $V_C^{\pi_o^{(t)}}(s_t)$. Using the common initial distribution and zero terminal values therefore yields:

$$\begin{aligned} &\mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t c_t\right] - J_C(\pi_o^{(0)}) \\ &\quad = \sum_{t=0}^{H-1} \gamma^t \mathbb{E}\left[Q_C^{\pi_o^{(t)}}(s_t, a_t) - Q_C^{\pi_o^{(t)}}(s_t, a_t^o)\right] \\ &\qquad + \sum_{t=1}^{H-1} \gamma^t \mathbb{E}\left[V_C^{\pi_o^{(t)}}(s_t) - V_C^{\pi_o^{(t-1)}}(s_t)\right]. \end{aligned} \tag{38}$$

The action-value difference is zero when the online candidate is selected. For a selected guide candidate, Eq. (29) applies to the current policy and estimator under the stated error condition. Applying this bound to the first sum and replacing each policy value difference in the second sum by its positive part proves Eq. (36). □

Even when policies and guide limits change within an episode, guide selection requires $t < h$. The guide contribution can therefore be bounded using the horizon sampled at episode start:

$$\begin{aligned} &\mathbb{E}\left[\sum_{t=0}^{H-1} I_t^g \left(2\gamma^t \varepsilon_C + \min\{\gamma^t \zeta, [b_t - \widehat{q}_t^o]_+\}\right)\right] \\ &\quad \le (\zeta + 2\varepsilon_C) \sum_{t=0}^{H-1} \gamma^t \Pr(h > t) \\ &\quad = (\zeta + 2\varepsilon_C) \sum_{t=0}^{H_{i_0}-1} \gamma^t \frac{H_{i_0} - t}{H_{i_0} + 1}. \end{aligned} \tag{39}$$

Taking positive parts in Eq. (36) then gives:

$$\begin{aligned} &\left[\mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t c_t\right] - d\right]_+ \\ &\quad \le [J_C(\pi_o^{(0)}) - d]_+ + (\zeta + 2\varepsilon_C) \sum_{t=0}^{H_{i_0}-1} \gamma^t \frac{H_{i_0} - t}{H_{i_0} + 1} \\ &\qquad + \sum_{t=1}^{H-1} \gamma^t \mathbb{E}\left[\left[V_C^{\pi_o^{(t)}}(s_t) - V_C^{\pi_o^{(t-1)}}(s_t)\right]_+\right]. \end{aligned} \tag{40}$$

The final term accounts for increases in cost value caused by policy updates. It vanishes when the policy is unchanged or each update satisfies $V_C^{\pi_o^{(t)}}(s_t) \le V_C^{\pi_o^{(t-1)}}(s_t)$ almost surely at the visited

state and episode step. In particular, under the fixed-policy conditions of Proposition 4.3, Eq. (40) reduces to its bound. For general updates, Eq. (36) guarantees that the expected episode cost does not exceed d if its right-hand side is at most d. Assumption 4.1 alone does not control the changes in policy value.

C.4 Discussion of Assumption 4.4

Assumption 4.4 gives conditions on the state distributions generated by RBGA roll-ins and the reward advantage of the learned policy. The distribution μ_t is induced by RBGA roll-ins, whose action selections may depend on the remaining budget, sampled horizon, and earlier observations. The value $Q_R^{\pi_o}$ evaluates the action at time t followed by the learned online policy π_o, as in guide-based policy search (Uchendu et al., 2023, Appendix A.5).

The state coverage condition in Eq. (12) compares probabilities of measurable sets. In a finite state space, it gives $\rho_t^{\pi^\star}(s) \leq D_\infty \mu_t(s)$ for every state. Under this condition, the expectation of any non-negative measurable function satisfies:

$$\int f(s)\, \rho_t^{\pi^\star}(\mathrm{d}s) \leq D_\infty \int f(s)\, \mu_t(\mathrm{d}s), \tag{41}$$

where f is a non-negative measurable function of the state. The constant D_∞ bounds how much more frequently the optimal feasible policy visits a set of states than the roll-in policy. Since the remaining budget affects action selection, the coverage condition is imposed on the distributions generated by the complete RBGA selection rule.

The advantage condition in Eq. (13) takes the positive part after averaging over the comparator's actions and before averaging over states. This gives the non-negative function to which the state coverage inequality applies. For the optimal feasible policy itself, the expected advantage satisfies:

$$\mathbb{E}_{a\sim\pi^\star(\cdot|s)} A_R^{\pi^\star}(s,a) = 0. \tag{42}$$

C.5 Proof and Discussion of Proposition 4.5

Assumption 4.4. For the learned online policy π_o, there exist a finite coverage coefficient $D_\infty \geq 1$ and finite reward advantage bounds $\varepsilon_{R,t} \geq 0$ such that, at every time step t, the state distributions satisfy:

$$\rho_t^{\pi^\star} \leq D_\infty \mu_t, \tag{43}$$

where μ_t is the marginal state distribution at time t induced by RBGA roll-ins, $\rho_t^{\pi^\star}$ is the corresponding distribution under $\pi^\star$, and D_∞ bounds their relative probabilities. Under the roll-in distribution, the expected positive advantage of the optimal feasible policy satisfies:

$$\mathbb{E}_{s\sim\mu_t}\left[\left[\mathbb{E}_{a\sim\pi^\star(\cdot|s)} A_R^{\pi_o}(s,a)\right]_+\right] \leq \varepsilon_{R,t}, \tag{44}$$

where $\varepsilon_{R,t}$ bounds the remaining positive reward advantage of $\pi^\star$ under the roll-in distribution.

Proposition 4.5. Under Assumption 4.4, the reward gap between the learned online policy π_o and the optimal feasible policy $\pi^\star$ satisfies:

$$J_R(\pi^\star) - J_R(\pi_o) \leq D_\infty \sum_{t=0}^{H-1} \gamma^t \varepsilon_{R,t}. \tag{45}$$

Proof. To compare the reward performance of $\pi^\star$ and π_o, we use the finite-horizon reward form of the performance difference lemma (Kakade & Langford, 2002, Lemma 6.1). With both policies starting from ρ_0, we obtain:

$$J_R(\pi^\star) - J_R(\pi_o) = \sum_{t=0}^{H-1} \gamma^t \mathbb{E}_{s\sim\rho_t^{\pi^\star}}\left[\mathbb{E}_{a\sim\pi^\star(\cdot|s)} A_R^{\pi_o}(s,a)\right]. \tag{46}$$

By replacing each state's expected advantage with its positive part and applying the two conditions in Assumption 4.4, we obtain:

$$\begin{aligned} J_R(\pi^\star) - J_R(\pi_o) &\leq \sum_{t=0}^{H-1} \gamma^t \mathbb{E}_{s\sim\rho_t^{\pi^\star}} \left[\left[\mathbb{E}_{a\sim\pi^\star(\cdot|s)} A_R^{\pi_o}(s,a) \right]_+ \right] \\ &\leq D_\infty \sum_{t=0}^{H-1} \gamma^t \mathbb{E}_{s\sim\mu_t} \left[\left[\mathbb{E}_{a\sim\pi^\star(\cdot|s)} A_R^{\pi_o}(s,a) \right]_+ \right] \\ &\leq D_\infty \sum_{t=0}^{H-1} \gamma^t \varepsilon_{R,t}, \end{aligned} \tag{47}$$

where the second inequality applies Eq. (43) to a non-negative function, and the last inequality follows from Eq. (44). Therefore, the reward gap satisfies Eq. (14). □

For the same learned policy satisfying $J_C(\pi_o) \leq d$, the optimality of $\pi^\star$ gives:

$$0 \leq J_R(\pi^\star) - J_R(\pi_o) \leq D_\infty \sum_{t=0}^{H-1} \gamma^t \varepsilon_{R,t}. \tag{48}$$

Thus, for a sequence of feasible learned policies with a common finite coverage constant and $\varepsilon_{R,t}$ tending to zero at every time step, the reward gap tends to zero.