AN END-TO-END OFFLINE RL APPROACH TO MULTI-TURN AGENTIC ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Training multi-turn agents from coarse trajectory-level feedback poses a fundamental credit-assignment problem, particularly when their interactions with users, tools, and environments are stochastic. In this paper, we propose *Stochastic Trajectory Bellman Residual Minimization* (Stochastic-TBRM), an end-to-end trajectory-based offline reinforcement learning approach to this problem: given static logs of agent trajectories and a single episodic alignment signal, learn a policy over the agent's sequential decision process. Under realizability, regularity, and coverage conditions, we establish a finite-sample policy guarantee for approximate minimization of the reduced trajectory objective. Controlled stochastic MDPs demonstrate value recovery and useful control from trajectory returns. Interactive-agent benchmarks evaluate deployment pipelines, whose warm-start and action-grounding components are distinguished from the effects of the learning objective.

1 INTRODUCTION

Large language models (LLMs) are increasingly deployed as interactive agents that act over extended multi-turn trajectories. Instead of producing a single answer to a single prompt, an agent may ask clarifying questions, call tools, inspect external states, revise its plan, and only later produce a final answer or execute an action (Schick et al., 2023; Li et al., 2025; Zhang et al., 2025; Chakraborty et al., 2025). In this regime, alignment is no longer only a property of an isolated response. It is a property of the entire trajectory: whether the agent gathers the right information, uses tools appropriately, avoids unsafe actions, adapts to new observations, and eventually satisfies the user's objective.

This shift creates a fundamental mismatch with the dominant post-training interface for LLM alignment. Standard RLHF and preference optimization methods typically attach feedback to a single prompt-response pair (Christiano et al., 2017; Ouyang et al., 2022; Rafailov et al., 2023). Multi-turn agents, however, generate sequences of interdependent decisions. The quality of an early action may only become clear several turns later, after the user responds, a tool returns information, or the environment changes. In practice, it is usually much easier to obtain one outcome label for the whole episode—such as task success, a holistic preference score, or a safety-verifier judgment—than to obtain reliable supervision after every turn. Dense process labels are costly, and for many agentic actions they are not even well-defined: asking a question, opening a webpage, delaying an answer, or trying a tool can be beneficial or harmful depending on its downstream consequences.

Recent work has recognized this difficulty and introduced additional structure into multi-turn alignment. Some methods extend preference optimization to complete conversations or trajectories, correcting for trajectory length and normalization effects (Shani et al., 2024; Shi et al., 2024). Other methods design turn-level rewards, process reward models, progress estimators, LLM critics, verifiers, or hierarchical decompositions to make long-horizon learning tractable (Zeng et al., 2025; Djuhera et al., 2026; Zhang et al., 2026). These approaches are powerful when the intermediate supervision is reliable. But they also shift part of the alignment problem into the design of the intermediate signal. Their scalability depends on whether one can afford dense annotation, trust a process reward model, define a valid verifier, or hand-design a decomposition that transfers across tasks.

This paper takes a more direct route: casting multi-turn agentic alignment as an end-to-end offline reinforcement learning problem from episodic rewards. The learner receives a static dataset of agent trajectories and a scalar outcome label for each trajectory. It does not receive periodic rewards after each transition, dense process annotations, online environment access, or hand-labeled stepwise

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.

credit. The goal is to learn a policy that improves the agent's sequential decisions directly from this outcome-only supervision.

The main challenge is that agentic trajectories are **stochastic**. User responses, tool outputs, web pages, simulators, and execution environments are not deterministic functions of the agent's previous action (Yao et al., 2022; Zhou et al., 2023; Xie et al., 2024). Recent trajectory-level Bellman residual methods show that outcome-labeled trajectories can support value-based offline optimization in deterministic or nearly deterministic language-model settings (Wang et al., 2025; Yuan et al., 2025). However, applying these objectives directly to stochastic transitions introduces the classical double-sampling bias (Baird, 1995; Dann et al., 2014; Feng et al., 2019). Intuitively, a single observed next state cannot distinguish whether a value prediction is wrong or whether the environment simply realized a noisy transition. This statistical issue is especially severe in offline agent logs, where we usually observe only one continuation for each state-action pair. This leads to the central question of this paper:

Can we learn effective multi-turn policies from only episodic outcome labels, even under stochastic transitions, without access to transition-level rewards or dense process supervision?

We answer this question affirmatively, both theoretically and empirically. Our main contributions are:

- **Episodic-reward offline RL for stochastic trajectories.** We formulate multi-turn agent alignment as an offline RL problem in which only a single trajectory-level outcome is observed. We introduce Stochastic-TBRM, which uses an auxiliary conditional-expectation critic to separate **stochastic** transition noise from Bellman error and fit trajectory-level outcomes from ordinary single-sample trajectories.
- **Theoretical guarantees.** We show that, at the population level, the auxiliary critic recovers the conditional expected continuation value. We establish a finite-sample policy guarantee for the reduced trajectory objective under realizability, regularity, and coverage conditions. For double stop-gradient updates, we give a best-iterate stationarity bound separating gradient noise, truncation bias, and auxiliary tracking error.
- **Validation on simulation and interactive agent benchmarks.** We evaluate Stochastic-TBRM on both controlled stochastic simulation benchmarks and interactive agent benchmarks. Controlled MDPs isolate learning from trajectory labels; interactive benchmarks evaluate complete deployment pipelines rather than isolate the benefit of critic updates.

2 PRELIMINARY

Notation. Let $\Delta_{\mathcal{A}}$ denote the probability simplex over a finite set $\mathcal{A}$, and let $\Delta_{\mathcal{A}}^{\mathcal{B}} := \{f : \mathcal{B} \to \Delta_{\mathcal{A}}\}$ denote the set of probability kernels. For a policy π, let $\mathbb{P}^{\pi}$ (resp. $\mathbb{E}^{\pi}$) denote the induced trajectory distribution (resp. expectation).

Environment and trajectories. We model a multi-turn conversational (or agentic) interaction as an episodic horizon-H MDP (possibly with early stopping): $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \nu_0, \gamma, r)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is a finite action space, $P(\cdot \mid s, a)$ is a stochastic transition kernel, ν_0 is the initial distribution, $\gamma \in (0, 1)$, and $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$. A state contains the decision index and full conversational context, including tool state and agent memory; actions are discrete high-level acts such as asking the user, calling a tool, or finalizing a response. The kernel captures stochastic user responses, tool outputs, and environment dynamics. We write $R(s, a) := r(s, a)$ below. All value functions are fixed to zero at termination. For horizon-H notation, early-stopped paths are padded with zero-reward absorbing steps, a single dummy action, and zero critic outputs.

A Markov policy $\pi \in \Delta_{\mathcal{A}}^{\mathcal{S}}$ maps each nonterminal state to an action distribution. The decision index in the state allows time-dependent behavior without a separate policy index. For a trajectory $\tau = (s_0, a_0, \ldots, s_H)$, its law factorizes as $\mathbb{P}^{\pi}(d\tau) = \nu_0(ds_0) \prod_{h=0}^{H-1} \pi(a_h \mid s_h)\, P(ds_{h+1} \mid s_h, a_h)$.

Offline data. We observe $\mathcal{D} = \{(\tau_i, y_i)\}_{i=1}^{n}$ generated by a behavior policy π_b. Each label is the exact discounted return, $y_i = R(\tau_i)$, where $R(\tau) := \sum_{h=0}^{H-1} \gamma^h R(s_h, a_h)$. This includes outcome-only supervision represented as a terminal reward. The learner observes trajectories and

their labels, but not the individual rewards $R(s_h, a_h)$. Let d_h^π be the law of (s_h, a_h) under π, and define $d^\pi := \frac{1}{C_H} \sum_{h=0}^{H-1} \gamma^h d_h^\pi$, where $C_H := \sum_{h=0}^{H-1} \gamma^h = \frac{1-\gamma^H}{1-\gamma}$.

Entropy-regularized objective. To allow randomized choices among macro actions, we adopt the standard soft RL objective (Ziebart et al., 2008; Ziebart, 2010; Neu et al., 2017),

$$J(\pi) := \mathbb{E}^\pi \left[\sum_{h=0}^{H-1} \gamma^h \big(R(s_h, a_h) - \beta \log \pi(a_h \mid s_h) \big) \right],$$

where $\beta > 0$ is the entropy coefficient throughout. Entropy regularization replaces the hard maximum in Bellman optimality with a smooth log-sum-exp operator and yields a unique softmax action distribution (Geist et al., 2019): Let $\pi^\star \in \operatorname{argmax}_\pi J(\pi)$. With time included in the state, the optimal remaining-horizon values satisfy (see Appendix B.4 in Kang et al. (2025))

$$\pi^\star(a \mid s) = \exp\left(\frac{Q^\star(s,a) - V^\star(s)}{\beta} \right), \qquad V^\star(s) = \beta \log \sum_{a \in \mathcal{A}} \exp\left(\frac{Q^\star(s,a)}{\beta} \right), \tag{1}$$

$$Q^\star(s,a) = R(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[V^\star(s')]. \tag{2}$$

Unlike hard greedy selection, this distribution varies smoothly with the action values, avoiding discontinuous policy changes at action-value ties. The resulting soft value computation is differentiable with respect to critic outputs, a property used in the reduced-objective optimization analysis below.

Remark 1 (Extension to KL regularization). This setting naturally extends to KL regularization against a fixed reference policy, as widely used in LLM alignment, including RLHF and the objective underlying DPO (Ouyang et al., 2022; Rafailov et al., 2023). For a reference policy π_{ref} with full support, define $R'(s,a) := R(s,a) + \beta \log \pi_{\text{ref}}(a \mid s)$. Then $R(s,a) - \beta \log \frac{\pi(a|s)}{\pi_{\text{ref}}(a|s)} = R'(s,a) - \beta \log \pi(a \mid s)$. Thus, the KL-regularized objective is exactly an entropy-regularized objective with the modified reward R'. For simplicity of exposition, we present the paper in the entropy-regularized form.

3 STOCHASTIC-TBRM

In this section, we introduce our main algorithm, **Stochastic Trajectory Bellman Residual Minimization (Stochastic-TBRM)**, for learning multi-turn agent policies from outcome-labeled offline trajectories under stochastic transitions.

For a critic Q, define its soft value and induced policy at nonterminal states as in Eq. (1); set $V_Q = 0$ at termination:

$$V_Q(s) := \beta \log \sum_{a \in \mathcal{A}} \exp\left(\frac{Q(s,a)}{\beta} \right), \qquad \pi_Q(a \mid s) = \frac{\exp\left(Q(s,a)/\beta\right)}{\sum_{a' \in \mathcal{A}} \exp\left(Q\left(s,a'\right)/\beta\right)}.$$

Bellman residual minimization The fixed-point identity $Q^\star = TQ^\star$ in Eq. (2) motivates fitting a critic by minimizing $Q - TQ$, a classical approach known as Bellman residual minimization (BRM) (Schweitzer & Seidmann, 1985; Baird, 1995; Antos et al., 2008). The objective is

$$\mathcal{L}_{\text{BRM}}(Q) := \mathbb{E}_{(s,a) \sim d^{\pi_b}} \left[\left((Q - TQ)(s,a) \right)^2 \right] \text{ where } (TQ)(s,a) := R(s,a) + \gamma \mathbb{E}[V_Q(s') \mid s, a].$$

Under stochastic transitions, however, replacing the expected continuation value with a single sampled value introduces a Q-dependent variance term. Direct unbiased estimation of the BRM gradient generally requires two conditionally independent next states for the same (s,a), the classical **double-sampling problem** (Baird, 1995; Jiang & Xie, 2025). Yuan et al. (2025) extend BRM to trajectory-level outcome supervision in LLMs, exploiting deterministic token-prefix transitions under which sampled and expected continuation values coincide. Our stochastic agentic setting does not admit this simplification. Instead, we introduce an **auxiliary function** $\xi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and fit it by regression (Antos et al., 2008; Dai et al., 2017; 2018):

$$\mathcal{J}(Q, \xi) \;:=\; \mathbb{E}_{\mathcal{D}} \Big[\big(V_Q(s') - \xi(s,a) \big)^2 \Big]. \tag{3}$$

Here $\mathbb{E}_{\mathcal{D}}$ averages over $(s,a) \sim d^{\pi_b}$ and $s' \sim P(\cdot \mid s, a)$. For fixed Q, the population regression target is the expected continuation value.

Lemma 1 (Best-response characterization of ξ)**.** *Fix a bounded Q and define $\xi^Q(s,a) := \mathbb{E}[V_Q(s') \mid s,a]$. If ξ ranges over all square-integrable measurable functions, or a class containing ξ^Q, every minimizer of $\xi \mapsto \mathcal{J}(Q,\xi)$ equals ξ^Q d^{π_b}-almost everywhere.*

Lemma 1 explains the role of ξ: it predicts expected continuation, not a hidden reward label. At its population best response, $Q - \gamma\xi^Q$ is a Bellman-implied reward proxy whose discounted sum can be fitted to episode returns.

Trajectory Bellman residual minimization. Unfortunately, a debiased BRM objective still requires the unobserved reward of each transition. Our data reveal only their discounted sum $R(\tau)$. Rather than prescribing a stepwise decomposition of this label, we match the accumulated reward implied by the critic to the observed outcome. Following the principle of Yuan et al. (2025), define

$$\widehat{R}_{Q,\xi}(s,a) := Q(s,a) - \gamma\xi(s,a), \qquad G_{Q,\xi}(\tau) := \sum_{h=0}^{H-1} \gamma^h \widehat{R}_{Q,\xi}(s_h,a_h). \tag{4}$$

The key change is to aggregate Bellman residuals before squaring, as proposed by Yuan et al. (2025). For any bounded Q, let $\delta_Q(s,a) := (TQ)(s,a) - Q(s,a)$ be the stepwise Bellman residual. The two population criteria are

$$\text{BRM: } \mathbb{E}_{\tau\sim\mathbb{P}^{\pi_b}}\left[\sum_{h=0}^{H-1}\gamma^h\delta_Q(s_h,a_h)^2\right] \quad \text{vs} \quad \text{TBRM: } \mathbb{E}_{\tau\sim\mathbb{P}^{\pi_b}}\left[\left(\sum_{h=0}^{H-1}\gamma^h\delta_Q(s_h,a_h)\right)^2\right]. \tag{5}$$

BRM penalizes each one-step residual separately, whereas TBRM penalizes their discounted sum. This change makes the unobserved rewards enter the loss only through $R(\tau)$, without prescribing their stepwise decomposition. Although the trajectory criterion permits cancellation across steps, $Q^\star$ still has zero loss, and Section 4.1 establishes its policy implications under the stated coverage conditions.

In the deterministic token-prefix setting of Yuan et al. (2025), the observed next-state value equals its conditional expectation, allowing the TBRM loss to be evaluated directly from outcome-labeled trajectories. On the other hand, our agentic setting includes stochastic transitions, so this substitution would generally change the objective. Instead, we carry the conditional-mean auxiliary from the BRM dual formulation into the trajectory criterion (Antos et al., 2008; Dai et al., 2018; Kang et al., 2025). At the population best response in Lemma 1, $G_{Q,\xi^Q}(\tau) - R(\tau)$ equals the summed residual in Eq. (5). We therefore use the following dual-inspired, auxiliary best-response formulation:

$$\min_{Q\in\mathcal{Q}} \mathbb{E}_{\tau\sim\mathbb{P}^{\pi_b}}\left[\big(G_{Q,\xi^Q}(\tau) - R(\tau)\big)^2\right], \quad \text{where} \quad \xi^Q \in \arg\min_{\xi} \mathcal{J}(Q,\xi). \tag{6}$$

The BRM dual transformation underlying this auxiliary construction is given in Appendix A.1.

3.1 Optimization and algorithm

The auxiliary formulation in Eq. (6) naturally leads to two empirical components: a trajectory-fitting loss $\mathcal{L}_{\text{fit}}$ and an auxiliary regression loss $\widehat{\mathcal{J}}$. For parametrized models Q_θ and ξ_ϕ, define

$$\widehat{\mathcal{L}}_{\text{fit}}(Q,\xi) := \frac{1}{n}\sum_{i=1}^{n}\big(G_{Q,\xi}(\tau_i) - y_i\big)^2, \quad \widehat{\mathcal{J}}(Q,\xi) := \frac{1}{N}\sum_{(s,a,s')\in\mathcal{D}_{\text{tr}}}\big(V_Q(s') - \xi(s,a)\big)^2. \tag{7}$$

where $\mathcal{D}_{\text{tr}}$ contains the N observed transitions and $y_i = R(\tau_i)$ for notational brevity. The first fits trajectory outcomes; the second tracks the conditional-mean target required by the first.

A standard approach to BRM's minimax formulation is alternating gradient ascent–descent, as used by Kang et al. (2025). Its auxiliary ascent step maximizes a negative squared-error term and is therefore implemented by minimizing the auxiliary regression loss. We adopt this blockwise learning structure for the trajectory formulation: with Q_θ fixed, we minimize $\lambda_\xi\widehat{\mathcal{J}}$ over ϕ; with ξ_ϕ fixed, we minimize $\mathcal{L}_{\text{fit}}$ over θ.

A double stop-gradient implements this separation. The auxiliary update treats $V_{Q_\theta}(s')$ as a fixed target, while the critic update treats $\xi_\phi(s,a)$ as fixed. These are alternating updates of two coupled

regression losses, rather than gradient ascent–descent on a common saddle objective. In particular, the critic step omits differentiation through the auxiliary best response; Section 4.2 analyzes the resulting bias. Algorithm 1 gives the minibatch procedure, with K_ξ auxiliary updates per critic update.

Algorithm 1 Stochastic TBRM (Double Stop-Gradient Variant)

Require: Offline dataset of labeled trajectories $\mathcal{D} = \{(\tau_i, y_i)\}_{i=1}^n$ with $y_i = R(\tau_i)$.
Require: Discount $\gamma \in (0,1)$, entropy parameter $\beta > 0$, auxiliary weight $\lambda_\xi > 0$, action set $\mathcal{A}$.
Require: Learning rates η_Q, η_ξ, batch sizes B_τ, B_{tr}, iterations T.
Require: Number of auxiliary updates per critic update K_ξ.
Require: $\mathrm{sg}(\cdot)$: stop-gradient; $m(s) = 0$ at termination and 1 otherwise.
Initialize parameters θ for $Q_\theta(s,a)$ and ϕ for $\xi_\phi(s,a)$.
Build a transition buffer $\mathcal{D}_{\text{tr}}$ from $\mathcal{D}$: $\mathcal{D}_{\text{tr}} = \{(s_h, a_h, s_{h+1}, h)$ from each $\tau_i\}$.
for $t = 1$ to T **do**
  for $k = 1$ to K_ξ **do**
    Sample a minibatch of transitions $\{(s^{(j)}, a^{(j)}, s'^{(j)}, h^{(j)})\}_{j=1}^{B_{\text{tr}}} \sim \mathcal{D}_{\text{tr}}$.
    for $j = 1$ to B_{tr} **do**
      $V^{(j)} \leftarrow \mathrm{sg}\Big(m(s'^{(j)})\,\beta \log \sum_{a\in\mathcal{A}} \exp\Big(Q_\theta(s'^{(j)}, a)/\beta\Big)\Big)$.
    end for
    $\mathcal{L}_\xi \leftarrow \frac{1}{B_{\text{tr}}} \sum_{j=1}^{B_{\text{tr}}} \Big(V^{(j)} - \xi_\phi(s^{(j)}, a^{(j)})\Big)^2$.
    $\phi \leftarrow \phi - \eta_\xi \nabla_\phi\big(\lambda_\xi\, \mathcal{L}_\xi\big)$.
  end for
  Sample a minibatch of trajectories $\{(\tau^{(i)}, y^{(i)})\}_{i=1}^{B_\tau} \sim \mathcal{D}$.
  for $i = 1$ to B_τ **do**
    $G^{(i)} \leftarrow 0$.
    for $h = 0$ to $\ell_i - 1$, where ℓ_i is the number of decisions in $\tau^{(i)}$ **do**
      Read (s_h, a_h) from $\tau^{(i)}$.
      $G^{(i)} \leftarrow G^{(i)} + \gamma^h \Big(Q_\theta(s_h, a_h) - \gamma\, \mathrm{sg}\big(\xi_\phi(s_h, a_h)\big)\Big)$.
    end for
  end for
  $\mathcal{L}_{\text{traj}} \leftarrow \frac{1}{B_\tau} \sum_{i=1}^{B_\tau} \Big(G^{(i)} - y^{(i)}\Big)^2$.
  $\theta \leftarrow \theta - \eta_Q \nabla_\theta \mathcal{L}_{\text{traj}}$.
end for
Return: Q_θ, ξ_ϕ and induced policy $\pi_\theta(a \mid s) = \exp\big((Q_\theta(s,a) - V_\theta(s))/\beta\big)$, where $V_\theta = V_{Q_\theta}$.

4 Theoretical Analysis

This section develops theoretical guarantees for the objective and critic updates in Algorithm 1, in terms of policy guarantee (Section 4.1) and optimization guarantee (Section 3.1).

4.1 Policy guarantee

The policy guarantee concerns approximate minimization of the reduced objective, in which $\xi = \xi^Q$ to the population target in Lemma 1. The reduced population loss and its empirical counterpart are

$$\mathcal{L}_{\mathrm{traj}}(Q) \;:=\; \mathbb{E}_{\tau\sim\mathbb{P}^{\pi_b}}\Big[\big(G_{Q,\xi^Q}(\tau) - R(\tau)\big)^2\Big], \qquad \widehat{\mathcal{L}}_{\mathrm{traj}}(Q) = \frac{1}{n}\sum_{i=1}^{n} \big(G_{Q,\xi^Q}(\tau_i) - R(\tau_i)\big)^2. \tag{8}$$

for independent trajectories $\tau_1, \ldots, \tau_n \sim \mathbb{P}^{\pi_b}$. The following identity connects the trajectory loss to the Bellman residual.

Lemma 2 (Trajectory residual equals stochastic trajectory Bellman residual)**.** *We have for every trajectory τ,*

$$G_{Q,\xi^Q}(\tau) - R(\tau) = -\sum_{h=0}^{H-1} \gamma^h \delta_Q(s_h, a_h), \quad \mathcal{L}_{\mathrm{traj}}(Q) = \mathbb{E}_{\tau\sim\mathbb{P}^{\pi_b}}\Big[\Big(\sum_{h=0}^{H-1} \gamma^h \delta_Q(s_h, a_h)\Big)^2\Big]. \tag{9}$$

The identity turns outcome-prediction error at the population auxiliary target into a discounted sum of Bellman residuals. These residuals can cancel across steps, so trajectory fitting need not recover each hidden reward. We instead relate this aggregate error directly to policy performance.

We use the standard realizability assumption from offline RL (Chen & Jiang, 2019; Xie & Jiang, 2021; Jiang & Xie, 2025) and boundedness and smoothness conditions for uniform convergence. For a parametrized critic Q_θ, write $\xi^\theta := \xi^{Q_\theta}$.

Assumption 1 (Realizability on the data support)**.** $Q^\star \in \mathcal{Q}$ and $\xi^\star(s,a) := \mathbb{E}[V^\star(s') \mid s,a]$ is representable (or ξ is unrestricted), at least for $(s,a) \in \mathrm{supp}(d^{\pi_b})$.

Assumption 2 (Bounded and Lipschitz reduced trajectory class)**.** For all $\theta, \theta' \in \Theta$ and (s,a), there are finite constants $B_Q, B_\xi, B_R, L_Q, L_\xi$ such that

$$|Q_\theta(s,a)| \le B_Q, \qquad |\xi^\theta(s,a)| \le B_\xi, \qquad |R(s,a)| \le B_R,$$

$$|Q_\theta(s,a) - Q_{\theta'}(s,a)| \le L_Q\|\theta - \theta'\|, \quad |\xi^\theta(s,a) - \xi^{\theta'}(s,a)| \le L_\xi\|\theta - \theta'\|.$$

Equip Θ with norm $\|\cdot\|$ and let $\mathcal{N}(\Theta, \alpha)$ denote its α-covering number, assumed finite at the radius.

The bound also depends on a *concentrability coefficient* measuring how far the occupancy measures of the target policies can deviate from the behavior policy, introduced by Jia et al. (2025); Yuan et al. (2025):

$$C_{\mathrm{conc}}(\pi_b) := 1 + \sup_{Q \in \mathcal{Q}} \sup_{\pi \in \{\pi^\star, \pi_Q\}} \sum_{h=0}^{H-1} \chi^2(d_h^\pi \| d_h^{\pi_b}).$$

Realizability supplies a zero-loss comparator; boundedness and covering control statistical error. $C_{\mathrm{conc}}(\pi_b)$ transfers behavior-distribution fit to the evaluated policies. It is a coverage condition, not a training penalty: poor coverage weakens or invalidates a finite bound (Appendix A.2).

Theorem 3 (Finite-sample policy guarantee)**.** *Suppose Assumptions 1 and 2 hold, and $C_{\mathrm{conc}}(\pi_b) < \infty$. Let $\widehat{Q} \in \mathcal{Q}$ be any $\varepsilon_{\mathrm{opt}}$-approximate empirical minimizer of the reduced trajectory loss, i.e.,*

$$\widehat{\mathcal{L}}_{\mathrm{traj}}(\widehat{Q}) \le \inf_{Q \in \mathcal{Q}} \widehat{\mathcal{L}}_{\mathrm{traj}}(Q) + \varepsilon_{\mathrm{opt}}.$$

Then for any $\alpha > 0$ with $\mathcal{N}(\Theta, \alpha) < \infty$ and $\delta \in (0,1)$, with probability at least $1 - \delta$,

$$J(\pi^\star) - J(\pi_{\widehat{Q}}) \le 2\sqrt{C_{\mathrm{conc}}(\pi_b)\big(2\varepsilon_{\mathrm{opt}} + \varepsilon_{\mathrm{stat}}(n, \alpha, \delta)\big)},$$

where

$$\varepsilon_{\mathrm{stat}}(n, \alpha, \delta) := \frac{10B^2}{3n} \log \frac{2\mathcal{N}(\Theta, \alpha)}{\delta} + 6BL\alpha, \qquad \begin{aligned} B &:= C_H(B_Q + \gamma B_\xi + B_R), \\ L &:= C_H(L_Q + \gamma L_\xi). \end{aligned}$$

In particular, under standard parametric entropy growth, $\log \mathcal{N}(\Theta, \alpha) \lesssim \dim(\Theta) \log(C/\alpha)$, choosing $\alpha = n^{-1}$ yields, for fixed horizon and regularity constants,

$$J(\pi^\star) - J(\pi_{\widehat{Q}}) \le \widetilde{O}\left(\sqrt{C_{\mathrm{conc}}(\pi_b)\Big(\varepsilon_{\mathrm{opt}} + \frac{\dim(\Theta)}{n}\Big)}\right).$$

Theorem 3 shows that outcome-only regression can control the loss of the induced soft policy without first recovering stepwise rewards. Its statistical contribution is $\widetilde{O}(\sqrt{C_{\mathrm{conc}} d/n})$, where $d = \dim(\Theta)$, for fixed horizon and regularity constants.

4.2 Optimization guarantee

The optimization analysis separately accounts for stop-gradient bias and imperfect auxiliary tracking. We analyze the plain-SGD critic updates in Algorithm 1 as biased stochastic gradients. Throughout this subsection and Appendix A.3, condition on the fixed dataset and initialization. Write $\widehat{\mathbb{E}}_{\mathcal{D}}$ for

the uniform average over stored trajectories; $\mathbb{E}$ denotes expectation over optimization randomness. Define

$$F(\theta) := \widehat{\mathcal{L}}_{\mathrm{traj}}(Q_\theta) = \widehat{\mathbb{E}}_{\mathcal{D}}\Big[\big(G_{Q_\theta,\xi^\theta}(\tau) - R(\tau)\big)^2\Big], \quad \widetilde{F}(\theta,\phi) := \widehat{\mathbb{E}}_{\mathcal{D}}\Big[\big(G_{Q_\theta,\xi_\phi}(\tau) - R(\tau)\big)^2\Big],$$
$$g_{\mathrm{DSG}}(\theta,\phi) := \nabla_\theta \widetilde{F}(\theta,\phi). \tag{10}$$

Here F is the reduced empirical target objective, $\widetilde{F}$ freezes the auxiliary model, and g_{DSG} is the mean minibatch update direction. Their discrepancy comes from omitting the derivative of ξ^θ and using an auxiliary model that has not reached this population target.

Assumption 3 (Smoothness and lower bound)**.** The reduced empirical objective F is L_F-smooth on a convex region containing all iterates and is bounded below by $F_{\inf} > -\infty$.

Assumption 4 (Differentiability and bounded critic Jacobian)**.** The map $\theta \mapsto Q_\theta(s,a)$ is differentiable, differentiation may be interchanged with the conditional expectation defining ξ^θ, and $\sup_{s,a} \|\nabla_\theta Q_\theta(s,a)\|_2 \le G_Q < \infty$ throughout this region.

Assumption 5 (Stochastic gradient estimator)**.** Let $\mathcal{F}_t$ contain the dataset and all randomness before the critic minibatch at iteration t, including (θ_t, ϕ_t). Uniform trajectory sampling gives

$$\mathbb{E}[\hat{g}_t \mid \mathcal{F}_t] = g_{\mathrm{DSG}}(\theta_t,\phi_t), \qquad \mathbb{E}[\|\hat{g}_t - g_{\mathrm{DSG}}(\theta_t,\phi_t)\|_2^2 \mid \mathcal{F}_t] \le \sigma^2$$

for some finite σ^2, where the second inequality is assumed.

Theorem 4 (Residual-dependent stationarity bound for double stop-gradient updates)**.** *Under Assumptions 3–5, let $\theta_{t+1} = \theta_t - \eta\hat{g}_t$ with $0 < \eta \le 1/(4L_F)$. Define*

$$\bar{\varepsilon}_{\mathrm{traj}} := \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[F(\theta_t)], \qquad \bar{\varepsilon}_{\mathrm{track}} := \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\|\xi_{\phi_t} - \xi^{\theta_t}\|_\infty^2]. \tag{11}$$

Then

$$\min_{0\le t<T} \mathbb{E}[\|\nabla F(\theta_t)\|_2^2] \le \frac{4(F(\theta_0) - F_{\inf})}{\eta T} + 2L_F\eta\sigma^2 + c_1\bar{\varepsilon}_{\mathrm{traj}} + c_2\bar{\varepsilon}_{\mathrm{track}}, \tag{12}$$

where $c_1 = 24\gamma^2 C_H^2 G_Q^2$ and $c_2 = 24\gamma^2 C_H^4 G_Q^2$.

Remark 2 (Technical distinction from deterministic TBRM.). The analysis of Yuan et al. (2025) relies on deterministic transitions to identify observed trajectory residuals with sums of Bellman residuals – in this case, $V_Q(s_{h+1}) = \mathbb{E}[V_Q(s_{h+1})|s_h, a_h]$. On the other hand, under stochastic dynamics, sampled continuation values instead leave transition noise even at $Q^\star$. Building on auxiliary formulations of BRM (Antos et al., 2008; Dai et al., 2018; Kang et al., 2025), we retain the conditional mean in the trajectory criterion: at the population auxiliary best response, Lemma 2 recovers the residual identity and preserves $Q^\star$ as a zero-loss comparator. This allows the existing performance-difference and change-of-measure arguments to apply to the reduced loss. On Section 4.2, freezing the critic-dependent auxiliary introduces gradient bias even under exact tracking. Proposition 1 separates this stop-gradient bias from auxiliary tracking error, yielding the residual-dependent stationarity bound in Theorem 4.

5 Experiments

Our experiments are organized around two distinct questions. First, in controlled stochastic MDPs where the learning signal can be inspected directly, does Stochastic-TBRM recover useful Markovian value structure from *trajectory-level* reward labels alone? Second, in the intended agentic-AI setting, can the same objective train a useful critic for agentic deployment from outcome-only ReAct logs? The first set of experiments should be read as mechanistic diagnostics for the theory. The second set evaluates the method in interactive language-agent benchmarks.

5.1 Controlled MDPs

These diagnostics isolate the reward interface. Stochastic-TBRM receives only the transition sequence and a scalar trajectory return per episode, never the per-transition reward that generated it. We use two environments that test different things: FrozenLake has an exact soft-optimal Q against which value recovery can be measured directly, and Acrobot has no such target, so it tests whether the recovered values support control.

Benchmarks. FrozenLake-8x8 is a tabular, slippery, sparse-reward environment with an exact soft value-iteration solution, so we report Q-MAE against that solution. Acrobot is a longer-horizon continuous-state benchmark without an exact target, so we evaluate learned policies by online rollout. Both use a fixed offline dataset mixing greedy and exploratory behavior, train for 50,000 iterations, and report 10 seeds (Appendix B.1).

Baselines. Four offline-RL methods receive per-transition reward. FQI (Ernst et al., 2005) regresses Q onto soft Bellman targets with temperature one, so its fixed point is the reference solution. CQL (Kumar et al., 2020) penalizes out-of-distribution actions, and Off-GLADIUS (Park et al.) minimizes the debiased squared Bellman error by alternating updates on Q and an auxiliary variable ξ. OptiDICE (Lee et al., 2021) optimizes a stationary-distribution correction without an explicit Q-function, so it has no FrozenLake Q-MAE. As an ablation, TBRM (Yuan et al., 2025) uses our trajectory-return interface but assumes deterministic transitions and omits ξ.

Table 1: Final values on the controlled stochastic MDPs, mean $\pm$ standard deviation over 10 seeds. FrozenLake reports Q-MAE against exact soft value iteration after 50,000 iterations, lower is better. Acrobot reports the best mean return logged during training, higher is better.

Reward input	Method	FrozenLake Q-MAE	Acrobot return
trajectory	TBRM	48.62 ± 0.03	-458.02 ± 98.61
trajectory	**Stochastic-TBRM (Ours)**	$\mathbf{2.10 \pm 0.63}$	$\mathbf{-87.04 \pm 3.70}$
transition	Off-GLADIUS	41.87 ± 0.02	-90.48 ± 8.01
transition	FQI	3.02 ± 0.18	-120.55 ± 29.90
transition	CQL	31.86 ± 0.14	-113.97 ± 37.14
transition	OptiDICE	–	-115.41 ± 14.70

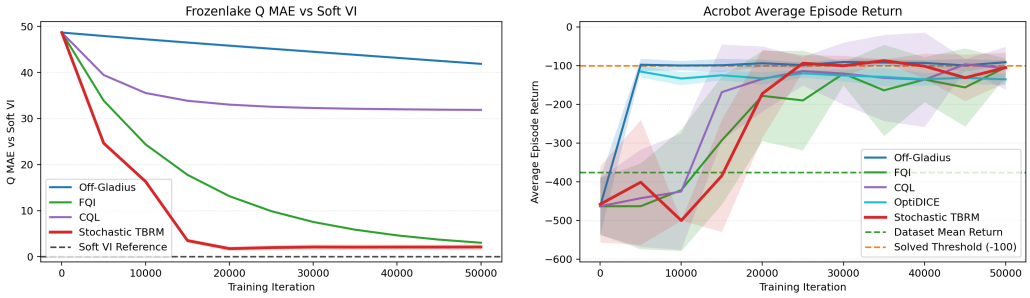

Figure 1: Learning curves on the controlled stochastic MDPs, mean over 10 seeds. Left: FrozenLake Q-MAE against exact soft value iteration. Right: Acrobot average episode return. Stochastic-TBRM is trained from trajectory-level returns only, and every baseline uses its standard transition-reward.

Results. Table 1 reports both benchmarks. On FrozenLake, TBRM without ξ stays near its initialization at 48.62, whereas Stochastic-TBRM reaches 2.10, comparable to transition-reward FQI from trajectory returns alone. On Acrobot, Stochastic-TBRM attains the highest return, and the same ablation again fails, confirming that the debiasing correction, not the trajectory interface, is what makes the objective work under stochastic transitions.

5.2 Interactive Agents

We next study interactive agents on WebShop, ScienceWorld, and ALFWorld, the three outcome-labeled benchmarks in the DMPO suite (Shi et al., 2024). We use the released environments, task splits, and offline ReAct demonstrations (Yao et al., 2022). Our deployment is value based, but its action grounding differs across benchmarks. On WebShop and ALFWorld, a text-conditioned critic scores benchmark-level macro actions before a fixed micro policy or deterministic executor produces a concrete command. On ScienceWorld, whose five macro actions are task phases, an exact-command policy proposes first and the critic intervenes only when a proposal is not executable.

Baselines. We compare against two policy-learning methods trained on the same demonstrations. DMPO (Shi et al., 2024) optimizes a multi-turn preference objective over trajectory pairs. Q-Evolve (Zhang et al., 2026) repeatedly collects its own rollouts, fits a critic on them together with the expert data under rule-based step penalties, and updates the policy with the resulting step-wise advantages. It therefore uses strictly richer supervision than the terminal scalar available to our critic, and it requires new environment interaction at every loop. We ran the official implementation under our matched contract with the same Qwen3.5-4B backbone and the authors' loop counts. On ALFWorld its performance is non-monotone across loops, and we report the prescribed loop rather than the stronger first loop (Appendix B.4).

Evaluation. All checkpoints were fixed on development data before test evaluation, and no policy reads simulator internals, answer keys, or held-out labels. ALFWorld is the only held-out benchmark, and it retains its behavior-cloning warm start because later Stochastic-TBRM iterates did not improve its development criterion. ScienceWorld is reported on a reused test because an earlier candidate failed its predeclared confirmation gate. WebShop is a locked evaluation rather than held-out evidence, because the released training index shares 21 identifiers with its 200-episode test index. Appendix B.4 gives the full contract.

Table 2: Interactive-agent results under a matched evaluation contract. "Completion" is full normalized score on ScienceWorld and task success otherwise. "Steps" is the mean number of executed commands per episode. Q-Evolve is reported at its authors' prescribed loop. Bold marks the best value within a benchmark, and lower steps are better.

Benchmark	Method	Episodes	Reward	Completion	Steps
WebShop	DMPO	200	0.649	0.955	**3.97**
	Q-Evolve	200	0.674	0.940	4.51
	Stochastic-TBRM (Ours)	200	**0.682**	**1.000**	4.03
ScienceWorld	DMPO	211	0.405	0.156	**19.56**
	Q-Evolve	211	0.520	0.265	21.95
	Stochastic-TBRM (Ours)	211	**0.539**	**0.308**	26.21
ALFWorld	DMPO	134	0.828	0.828	19.07
	Q-Evolve	134	0.463	0.463	25.71
	Stochastic-TBRM (Ours)	134	**1.000**	**1.000**	**10.01**

Results. Table 2 reports the outcomes. Our method completes every ALFWorld task with the fewest steps and attains the highest reward on the other two benchmarks. On ScienceWorld it performs comparably to Q-Evolve from the terminal scalar alone, whereas Q-Evolve additionally receives step-level penalties and repeated environment interaction. These benchmarks evaluate complete deployment pipelines rather than the critic in isolation. ALFWorld retains its BC initialization and ScienceWorld's BC-shield variant also improves over DMPO, so neither result separates the contribution of Stochastic-TBRM updates from the action-grounding interface (Table 6).

6 Conclusion

We introduced *Stochastic Trajectory Bellman Residual Minimization* (Stochastic-TBRM) for learning from offline multi-turn trajectories under stochastic transitions and outcome-only supervision. An auxiliary conditional-expectation model removes the sampled-next-state substitution at its population best response. We established a finite-sample policy guarantee for the reduced trajectory objective and a residual-dependent stationarity bound for double stop-gradient updates; these do not constitute an end-to-end guarantee for the fitted auxiliary model and optimizer. Controlled stochastic MDPs demonstrate value recovery and useful control from trajectory labels. Interactive benchmarks assess complete deployment pipelines: the ALFWorld BC checkpoint solves all 134 held-out tasks, while ScienceWorld and WebShop results depend on benchmark-specific grounding and their reported evaluation contracts. These results do not isolate the benefit of critic updates. Future work should connect auxiliary estimation to policy guarantees and study partial coverage, weaker observability, and scalable command grounding.

Reproducibility Statement

We provide an anonymous implementation of Stochastic-TBRM and the scripts used for the reported experiments at `https://anonymous.4open.science/r/stochastic-TBRM-BA04`. The main algorithm is specified in Algorithm 1, and the assumptions and complete proofs for the theoretical results are provided in Appendix A. Appendix B documents the experimental setup in detail, including benchmark environments and data construction, information boundaries, model and optimization hyperparameters, random seeds, checkpoint-selection procedures, evaluation protocols, and reproducibility identifiers for the reported runs. The accompanying code repository provides the corresponding training and evaluation code, configuration files, and run manifests needed to reproduce the reported tables and figures. Together, these materials are intended to enable independent reproduction of both the theoretical and empirical results.

AI Use Disclosure

In this work, we used generative AI tools to assist in the writing of proofs, propose or refine hypotheses, and assist with translation. We have not used generative AI tools for the other research tasks such as generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, design or provide feedback on research methodology or experiments, implement methods, clean and reformat dataset and support qualitative and thematic data analysis, interpret results. Additionally, we used generative AI tools to create or edit software code, draft parts of the research paper, summarize or analyze existing literature, and edit the paper to improve readability.

We have reviewed all AI-assisted work. In particular, we checked AI-assisted research ideas and literature analyses for potential plagiarism and unsupported attribution through a manual literature survey. AI-generated or AI-edited experimental code was verified and tested for correctness by at least two authors. All AI-assisted proofs and mathematical arguments were independently reviewed and verified by the authors, and all participating authors are fully familiar with the final proofs and experimental implementation.

We take responsibility for the final content of this work, including text, claims, code, or other artifacts produced with the aid of generative AI.

References

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Rémi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*, 2018.

András Antos, Csaba Szepesvári, and Rémi Munos. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71 (1):89–129, 2008.

Ashutosh Baheti, Ximing Lu, Faeze Brahman, Ronan Le Bras, Maarten Sap, and Mark Riedl. Leftover lunch: Advantage-based offline reinforcement learning for language models. *arXiv preprint arXiv:2305.14718*, 2023.

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 30–37, 1995.

Amartya Chakraborty, Paresh Dashore, Nadia Bathaee, Anmol Jain, Anirban Das, Shi-Xiong Zhang, Sambit Sahu, Milind R. Naphade, and Genta Winata. T1: A tool-oriented conversational dataset for multi-turn agentic planning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3479.

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International conference on machine learning*, pp. 1042–1051. PMLR, 2019.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Pierre Clavier, Nathan Grinsztajn, Raphael Avalos, Yannis Flet-Berliac, Irem Ergun, Omar D Domingues, Eugene Tarassov, Olivier Pietquin, Pierre H Richemond, Florian Strub, et al. Shiq: Bringing back bellman to llms. *arXiv preprint arXiv:2505.11081*, 2025.

Bo Dai, Niao He, Yunpeng Pan, Byron Boots, and Le Song. Learning from conditional distributions via dual embeddings. In Aarti Singh and Jerry Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1458–1467. PMLR, 2017. URL https://proceedings.mlr.press/v54/dai17a.html.

Bo Dai, Albert Shaw, Lihong Li, Lin Xiao, Niao He, Zhen Liu, Jianshu Chen, and Le Song. Sbeed: Convergent reinforcement learning with nonlinear function approximation. In *International conference on machine learning*, pp. 1125–1134. PMLR, 2018.

Christoph Dann, Gerhard Neumann, and Jan Peters. Policy evaluation with temporal differences: A survey and comparison. *Journal of Machine Learning Research*, 15:809–883, 2014.

Aladin Djuhera, Swanand Ravindra Kadhe, Farhan Ahmed, and Holger Boche. Tsr: Trajectory-search rollouts for multi-turn rl of llm agents. *arXiv preprint arXiv:2602.11767*, 2026.

Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6, 2005.

Yihao Feng, Lihong Li, and Qiang Liu. A kernel loss for solving the bellman equation. *Advances in Neural Information Processing Systems*, 32, 2019.

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

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized Markov decision processes. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2160–2169. PMLR, 2019.

Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*, 2022.

Xiang Ji, Sanjeev Kulkarni, Mengdi Wang, and Tengyang Xie. Self-play with adversarial critic: Provable and scalable offline alignment for language models. *arXiv preprint arXiv:2406.04274*, 2024.

Zeyu Jia, Alexander Rakhlin, and Tengyang Xie. Do we need to verify step by step? rethinking process supervision from a theoretical perspective. *arXiv preprint arXiv:2502.10581*, 2025.

Nan Jiang and Tengyang Xie. Offline reinforcement learning in large state spaces: Algorithms and guarantees. *Statistical Science*, 40(4):570–596, 2025.

Enoch Hyunwook Kang, Hema Yoganarasimhan, and Lalit Jain. An empirical risk minimization approach for offline inverse reinforcement learning and dynamic discrete choice models. In *Proceedings of the 26th ACM Conference on Economics and Computation*, pp. 341–341, 2025.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

Jongmin Lee, Wonseok Jeon, Byungjun Lee, Joelle Pineau, and Kee-Eung Kim. Optidice: Offline policy optimization via stationary distribution correction estimation. In *International Conference on Machine Learning*, pp. 6120–6130. PMLR, 2021.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Yubo Li, Xiaobin Shen, Xinyu Yao, Xueying Ding, Yidi Miao, Ramayya Krishnan, and Rema Padman. Beyond single-turn: A survey on multi-turn interactions with large language models. *arXiv preprint arXiv:2504.04717*, 2025.

Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Byungjun Park, Minhyeok Park, Enoch H Kang, and Kyoungseok Jang. Globally convergent offline reinforcement learning with smoothed bellman residual minimization. In *Decision-Making from Offline Datasets to Online Adaptation: Black-Box Optimization to Reinforcement Learning*.

Jiangweizhi Peng, Yuanxin Liu, Ruida Zhou, Charles Fleming, Zhaoran Wang, Alfredo Garcia, and Mingyi Hong. Hiper: Hierarchical reinforcement learning with explicit credit assignment for large language model agents. *arXiv preprint arXiv:2602.16165*, 2026.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

Paul J. Schweitzer and Abraham Seidmann. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110(2):568–582, 1985. doi: 10.1016/0022-247X(85)90317-8. URL https://www.sciencedirect.com/science/article/pii/0022247X85903178.

Lior Shani, Aviv Rosenberg, Asaf Cassel, Oran Lang, Daniele Calandriello, Avital Zipori, Hila Noga, Orgad Keller, Bilal Piot, Idan Szpektor, et al. Multi-turn reinforcement learning with preference human feedback. *Advances in Neural Information Processing Systems*, 37:118953–118993, 2024.

Wentao Shi, Mengqi Yuan, Junkang Wu, Qifan Wang, and Fuli Feng. Direct multi-turn preference optimization for language agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2312–2324, 2024.

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

Huaijie Wang, Shibo Hao, Hanze Dong, Shenao Zhang, Yilin Bao, Ziran Yang, and Yi Wu. Offline reinforcement learning for llm multi-step reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 8881–8893, 2025.

Tengyang Xie and Nan Jiang. Batch value-function approximation with only realizability. In *International Conference on Machine Learning*, pp. 11404–11413. PMLR, 2021.

Tianbao Xie et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *arXiv preprint arXiv:2404.07972*, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Yurun Yuan, Fan Chen, Zeyu Jia, Alexander Rakhlin, and Tengyang Xie. Trajectory bellman residual minimization: A simple value-based method for llm reasoning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Siliang Zeng, Quan Wei, William Brown, Oana Frunza, Yuriy Nevmyvaka, Yang Katie Zhao, and Mingyi Hong. Reinforcing multi-turn reasoning in llm agents via turn-level credit assignment. In *ICML 2025 Workshop on Computer Use Agents*, 2025.

Chen Zhang, Xinyi Dai, Yaxiong Wu, Qu Yang, Yasheng Wang, Ruiming Tang, and Yong Liu. A survey on multi-turn interaction capabilities of large language models. *arXiv preprint arXiv:2501.09959*, 2025.

Yudi Zhang, Meng Fang, Zhenfang Chen, and Mykola Pechenizkiy. Self-evolving llm agents with in-distribution optimization. *arXiv preprint arXiv:2606.07367*, 2026.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

Yuhua Zhu and Lexing Ying. Borrowing from the future: An attempt to address double sampling. In *Mathematical and scientific machine learning*, pp. 246–268. PMLR, 2020.

Brian D. Ziebart. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. PhD thesis, Carnegie Mellon University, 2010.

Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1433–1438, 2008.

A PROOFS

A.1 AUXILIARY CRITIC

Proof of Lemma 1. Conditional variance decomposition gives

$$\mathcal{J}(Q,\xi) = \mathbb{E}_{(s,a)\sim d^{\pi_b}}\left[\mathrm{Var}(V_Q(s') \mid s,a) + (\xi(s,a) - \xi^Q(s,a))^2\right].$$

The first term does not depend on ξ. When the class contains ξ^Q, the second term has minimum zero, so every minimizer equals ξ^Q d^{π_b}-almost everywhere. □

Connection to Bellman residual minimization. Let $L_{\mathrm{TD}}(Q)(s,a,s') := (R(s,a) + \gamma V_Q(s') - Q(s,a))^2$. At the auxiliary best response, $\mathcal{J}(Q,\xi^Q) = \mathbb{E}[\mathrm{Var}(V_Q(s') \mid s,a)]$. The variance-corrected BRM identity (Dai et al., 2018; Kang et al., 2025) is therefore

$$\max_{\xi}\left\{\mathbb{E}_{\mathcal{D}}[L_{\mathrm{TD}}(Q)] - \gamma^2\mathcal{J}(Q,\xi)\right\} = \mathbb{E}_{(s,a)\sim d^{\pi_b}}\left[(R(s,a) + \gamma\xi^Q(s,a) - Q(s,a))^2\right].$$

For fixed (Q,ξ), the expression on the left admits single-next-state estimates when stepwise rewards are available. This population identity motivates the conditional-mean model; it is not Algorithm 1's training objective. In particular, minimizing the sum of the two fitting losses with respect to Q would also differentiate the generally Q-dependent quantity $\mathcal{J}(Q,\xi^Q)$. The double stop-gradient updates do not take that derivative.

Remark 3 (Reference-policy regularization). A KL penalty to a known reference policy replaces the entropy-regularized reward by $R_{\mathrm{ref}}(s,a) := r(s,a) + \beta\log\pi_{\mathrm{ref}}(a \mid s)$ (Ouyang et al., 2022; Abdolmaleki et al., 2018). The corresponding trajectory label is obtained by adding $\beta\sum_h \gamma^h \log\pi_{\mathrm{ref}}(a_h \mid s_h)$ to the observed return. The same analysis applies to the shifted problem provided its realizability and regularity conditions hold and $\log\pi_{\mathrm{ref}}$ is uniformly bounded on the relevant state-action domain (for example, $\pi_{\mathrm{ref}} \geq p_{\min} > 0$ there). The reward and value bounds must be adjusted; positivity or boundedness cannot be omitted. The main text uses the reference-free formulation.

A.2 POLICY GUARANTEE

Proof of Lemma 2. Expand and substitute $\xi^Q(s_h,a_h) = \mathbb{E}[V_Q(s_{h+1}) \mid s_h,a_h]$:

$$\begin{aligned}
G_{Q,\xi^Q}(\tau) - R(\tau) &= \sum_{h=0}^{H-1}\gamma^h\Big(Q(s_h,a_h) - \gamma\xi^Q(s_h,a_h) - R(s_h,a_h)\Big)\\
&= -\sum_{h=0}^{H-1}\gamma^h\Big(R(s_h,a_h) + \gamma\mathbb{E}[V_Q(s_{h+1}) \mid s_h,a_h] - Q(s_h,a_h)\Big)\\
&= -\sum_{h=0}^{H-1}\gamma^h\big((TQ)(s_h,a_h) - Q(s_h,a_h)\big).
\end{aligned}$$

□

Lemma 5 (Soft performance difference via a reference function)**.** *Let π and $\pi^\dagger$ be policies, and let f_h be bounded measurable functions for $h = 0,\dots,H-1$, vanishing on terminal padding states. Define*

$$U_h(s) := \mathbb{E}_{a\sim\pi(\cdot|s)}[f_h(s,a) - \beta\log\pi(a \mid s)], \qquad U_H \equiv 0,$$

and $(T_h^\pi f_{h+1})(s,a) := R(s,a) + \gamma\mathbb{E}[U_{h+1}(s') \mid s,a]$, with all continuation terms zero at early termination. Then

$$\begin{aligned}
J(\pi^\dagger) - J(\pi) = \mathbb{E}^{\pi^\dagger}&\left[\sum_{h=0}^{H-1}\gamma^h\big((T_h^\pi f_{h+1})(s_h,a_h) - f_h(s_h,a_h)\big)\right]\\
&+ \mathbb{E}^{\pi}\left[\sum_{h=0}^{H-1}\gamma^h\big(f_h(s_h,a_h) - (T_h^\pi f_{h+1})(s_h,a_h)\big)\right]\\
&+ \mathbb{E}^{\pi^\dagger}\left[\sum_{h=0}^{H-1}\gamma^h\big(f_h(s_h,a_h) - \beta\log\pi^\dagger(a_h \mid s_h) - U_h(s_h)\big)\right].
\end{aligned}$$

Proof. Let $e_h := T_h^\pi f_{h+1} - f_h$. Conditional expectation and $U_H = 0$ give

$$\mathbb{E}^\pi\left[\sum_{h=0}^{H-1}\gamma^h e_h(s_h,a_h)\right] = J(\pi) - \mathbb{E}_{s_0\sim\nu_0}[U_0(s_0)].$$

For $\pi^\dagger$, the same telescoping calculation, after adding and subtracting its entropy term, yields

$$\mathbb{E}^{\pi^\dagger}\left[\sum_{h=0}^{H-1}\gamma^h e_h(s_h,a_h)\right] = J(\pi^\dagger) - \mathbb{E}_{s_0\sim\nu_0}[U_0(s_0)] \\ - \mathbb{E}^{\pi^\dagger}\left[\sum_{h=0}^{H-1}\gamma^h\big(f_h(s_h,a_h) - \beta\log\pi^\dagger(a_h\mid s_h) - U_h(s_h)\big)\right].$$

Subtracting the first identity from the second and rearranging proves the claim. □

Corollary 1 (Suboptimality controlled by the debiased trajectory Bellman residual)**.** *Let $Q : \mathcal{S}\times\mathcal{A}\to\mathbb{R}$ be bounded, with the terminal convention of Section 2. At nonterminal states define*

$$\pi_Q(a\mid s)\propto\exp\!\left(\frac{Q(s,a)}{\beta}\right), \qquad V_Q(s) := \beta\log\sum_{a\in\mathcal{A}}\exp\!\left(\frac{Q(s,a)}{\beta}\right).$$

Let

$$\xi^Q(s,a) := \mathbb{E}[V_Q(s')\mid s,a],$$

and define the debiased trajectory return

$$G_{Q,\xi^Q}(\tau) := \sum_{h=0}^{H-1}\gamma^h\big(Q(s_h,a_h) - \gamma\xi^Q(s_h,a_h)\big).$$

Then

$$J(\pi^\star) - J(\pi_Q) \le 2\max_{\pi\in\{\pi^\star,\pi_Q\}}\left|\mathbb{E}_{\tau\sim P^\pi}\left[G_{Q,\xi^Q}(\tau) - R(\tau)\right]\right|.$$

Equivalently, since

$$G_{Q,\xi^Q}(\tau) - R(\tau) = -\sum_{h=0}^{H-1}\gamma^h\big((TQ)(s_h,a_h) - Q(s_h,a_h)\big),$$

we also have

$$J(\pi^\star) - J(\pi_Q) \le 2\max_{\pi\in\{\pi^\star,\pi_Q\}}\left|\mathbb{E}_{\tau\sim P^\pi}\left[\sum_{h=0}^{H-1}\gamma^h\big((TQ)(s_h,a_h) - Q(s_h,a_h)\big)\right]\right|.$$

Proof. Apply Lemma 5 with $\pi^\dagger = \pi^\star$, $\pi = \pi_Q$, and $f_h = Q$ on the time-h state layer, with zero terminal continuation. By definition of $\pi_Q(a\mid s)\propto\exp(Q(s,a)/\beta)$, we have

$$V_Q(s) = \mathbb{E}_{a\sim\pi_Q(\cdot|s)}\big[Q(s,a) - \beta\log\pi_Q(a\mid s)\big].$$

Hence, for every (s,a),

$$(TQ)(s,a) = R(s,a) + \gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}[V_Q(s')] \\ = R(s,a) + \gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}\Big[\mathbb{E}_{a'\sim\pi_Q(\cdot|s')}\big(Q(s',a') - \beta\log\pi_Q(a'\mid s')\big)\Big].$$

Therefore, in the notation of Lemma 5,

$$T_h^{\pi_Q}Q = TQ \qquad \text{for all } h.$$

Moreover, since π_Q is the softmax policy induced by Q, it maximizes $\nu\mapsto\mathbb{E}_{a\sim\nu(\cdot|s)}[Q(s,a) - \beta\log\nu(a\mid s)]$ for every s. Hence

$$\mathbb{E}_{a\sim\pi^\star(\cdot|s)}\big[Q(s,a) - \beta\log\pi^\star(a\mid s)\big] - \mathbb{E}_{a\sim\pi_Q(\cdot|s)}\big[Q(s,a) - \beta\log\pi_Q(a\mid s)\big] \le 0$$

for every s. Thus the third term in Lemma 5 is nonpositive when $\pi^\dagger = \pi^\star$, $\pi = \pi_Q$, and $f_h = Q$. Applying Lemma 5 yields

$$J(\pi^\star) - J(\pi_Q) \le \mathbb{E}_{\pi^\star}\left[\sum_{h=0}^{H-1} \gamma^h \big((TQ)(s_h,a_h) - Q(s_h,a_h)\big)\right] + \mathbb{E}_{\pi_Q}\left[\sum_{h=0}^{H-1} \gamma^h \big(Q(s_h,a_h) - (TQ)(s_h,a_h)\big)\right].$$

Hence

$$J(\pi^\star) - J(\pi_Q) \le \left|\mathbb{E}_{\pi^\star}\left[\sum_{h=0}^{H-1} \gamma^h \big((TQ)(s_h,a_h) - Q(s_h,a_h)\big)\right]\right| + \left|\mathbb{E}_{\pi_Q}\left[\sum_{h=0}^{H-1} \gamma^h \big((TQ)(s_h,a_h) - Q(s_h,a_h)\big)\right]\right|.$$

Using $|x| + |y| \le 2\max\{|x|,|y|\}$, we obtain

$$J(\pi^\star) - J(\pi_Q) \le 2 \max_{\pi \in \{\pi^\star, \pi_Q\}} \left|\mathbb{E}_{\tau\sim P^\pi}\left[\sum_{h=0}^{H-1} \gamma^h \big((TQ)(s_h,a_h) - Q(s_h,a_h)\big)\right]\right|.$$

Finally, by Lemma 2 and the definition $\xi^Q(s,a) = \mathbb{E}[V_Q(s') \mid s,a]$, we have

$$(TQ)(s,a) = R(s,a) + \gamma \xi^Q(s,a),$$

so

$$G_{Q,\xi^Q}(\tau) - R(\tau) = \sum_{h=0}^{H-1} \gamma^h \big(Q(s_h,a_h) - \gamma\xi^Q(s_h,a_h) - R(s_h,a_h)\big) = -\sum_{h=0}^{H-1} \gamma^h \big((TQ)(s_h,a_h) - Q(s_h,a_h)\big).$$

Substituting this identity into the previous display gives the first bound as well. □

Lemma 6 (Change-of-trajectory-measure)**.** *Let π and μ be two Markov policies on an episodic horizon-H MDP, and let $d_h^\pi, d_h^\mu \in \Delta(\mathcal{S}\times\mathcal{A})$ denote their stage-h occupancy measures. Then*

$$\sup_f \frac{\left(\mathbb{E}_{\tau\sim P^\pi}\left[\sum_{h=0}^{H-1} f_h(s_h,a_h)\right]\right)^2}{\mathbb{E}_{\tau\sim P^\mu}\left[\left(\sum_{h=0}^{H-1} f_h(s_h,a_h)\right)^2\right]} \;\le\; 1 + \sum_{h=0}^{H-1} \chi^2(d_h^\pi \,\|\, d_h^\mu),$$

where the supremum is over measurable collections with a finite, positive denominator. The corresponding unnormalized inequality also covers a zero denominator when the right-hand concentrability coefficient is finite.

This is the change-of-trajectory-measure result of Yuan et al. (2025, Lemma 1), with the decision index included in the state. Applying it with $f_h(s,a) = \gamma^h \delta_Q(s,a)$ gives Corollary 2, which transfers behavior-distribution trajectory error to target-policy error.

Corollary 2. *For any bounded Q, letting*

$$\delta_Q(s,a) := (TQ)(s,a) - Q(s,a),$$

we have

$$\left(\mathbb{E}_{\tau\sim P^\pi}\left[\sum_{h=0}^{H-1} \gamma^h \delta_Q(s_h,a_h)\right]\right)^2 \le \left(1 + \sum_{h=0}^{H-1} \chi^2(d_h^\pi \| d_h^{\pi_b})\right) \mathbb{E}_{\tau\sim P^{\pi_b}}\left[\left(\sum_{h=0}^{H-1} \gamma^h \delta_Q(s_h,a_h)\right)^2\right].$$

Equivalently, by Lemma 2,

$$\big(\mathbb{E}_{\tau\sim P^\pi}\big[G_{Q,\xi^Q}(\tau) - R(\tau)\big]\big)^2 \le \left(1 + \sum_{h=0}^{H-1} \chi^2(d_h^\pi \| d_h^{\pi_b})\right) \mathcal{L}_{\mathrm{traj}}(Q).$$

Proof. Apply Lemma 6 with $\mu = \pi_b$ and with the measurable stage-wise functions

$$f_h(s,a) := \gamma^h \delta_Q(s,a) \qquad (h = 0, \dots, H-1).$$

Then

$$\left(\mathbb{E}_{\tau\sim P^\pi}\left[\sum_{h=0}^{H-1}\gamma^h\delta_Q(s_h,a_h)\right]\right)^2 \le \left(1+\sum_{h=0}^{H-1}\chi^2(d_h^\pi\|d_h^{\pi_b})\right)\mathbb{E}_{\tau\sim P^{\pi_b}}\left[\left(\sum_{h=0}^{H-1}\gamma^h\delta_Q(s_h,a_h)\right)^2\right].$$

This proves the first claim.

For the second claim, Lemma 2 states that for every trajectory τ,

$$G_{Q,\xi^Q}(\tau) - R(\tau) = -\sum_{h=0}^{H-1}\gamma^h\big((TQ)(s_h,a_h) - Q(s_h,a_h)\big) = -\sum_{h=0}^{H-1}\gamma^h\delta_Q(s_h,a_h).$$

Therefore,

$$\big(\mathbb{E}_{\tau\sim P^\pi}\big[G_{Q,\xi^Q}(\tau) - R(\tau)\big]\big)^2 = \left(\mathbb{E}_{\tau\sim P^\pi}\left[\sum_{h=0}^{H-1}\gamma^h\delta_Q(s_h,a_h)\right]\right)^2,$$

and

$$\mathcal{L}_{\mathrm{traj}}(Q) = \mathbb{E}_{\tau\sim P^{\pi_b}}\left[\big(G_{Q,\xi^Q}(\tau) - R(\tau)\big)^2\right] = \mathbb{E}_{\tau\sim P^{\pi_b}}\left[\left(\sum_{h=0}^{H-1}\gamma^h\delta_Q(s_h,a_h)\right)^2\right].$$

Substituting these identities into the first bound yields the second claim. □

Lemma 7 (Bernstein bound for the reduced trajectory loss)**.** *Let $\tau_1, \dots, \tau_n$ be independent trajectories drawn from $\mathbb{P}^{\pi_b}$ and set*

$$f_\theta(\tau) := G_{Q_\theta,\xi^\theta}(\tau) - R(\tau), \qquad \widehat{\mathcal{L}}_{\mathrm{traj}}(\theta) := \frac{1}{n}\sum_{i=1}^{n} f_\theta(\tau_i)^2.$$

Under Assumption 2, for any fixed $\alpha > 0$ with $\mathcal{N}(\Theta,\alpha) < \infty$ and $\delta \in (0,1)$, with probability at least $1-\delta$, uniformly over $\theta \in \Theta$,

$$\mathcal{L}_{\mathrm{traj}}(Q_\theta) \le 2\widehat{\mathcal{L}}_{\mathrm{traj}}(\theta) + \varepsilon_{\mathrm{stat}}(n,\alpha,\delta),$$

where

$$\varepsilon_{\mathrm{stat}}(n,\alpha,\delta) = \frac{10B^2}{3n}\log\frac{2\mathcal{N}(\Theta,\alpha)}{\delta} + 6BL\alpha,$$

$B = C_H(B_Q + \gamma B_\xi + B_R)$ and $L = C_H(L_Q + \gamma L_\xi)$.

Proof. Boundedness gives

$$|f_\theta(\tau)| \le \sum_{h=0}^{H-1}\gamma^h(B_Q + \gamma B_\xi + B_R) = B. \tag{13}$$

The global Lipschitz conditions in Assumption 2 imply

$$\|f_\theta - f_{\theta'}\|_\infty \le \sum_{h=0}^{H-1}\gamma^h(L_Q + \gamma L_\xi)\|\theta - \theta'\| = L\|\theta - \theta'\|. \tag{14}$$

Let $\Theta_\alpha = \{\bar\theta_1, \dots, \bar\theta_M\}$ be an α-net, where $M = \mathcal{N}(\Theta,\alpha)$, and let $u = \log(2M/\delta)$. For each net point, write $Z_i^{(j)} = f_{\bar\theta_j}(\tau_i)^2$ and $m_j = \mathbb{E}[Z_i^{(j)}]$. Since $0 \le Z_i^{(j)} \le B^2$,

$$\mathrm{Var}(Z_i^{(j)}) \le \mathbb{E}[(Z_i^{(j)})^2] \le B^2 m_j.$$

Bernstein's lower-tail inequality and a union bound imply that, with probability at least $1-\delta$, simultaneously for every j,

$$\begin{aligned} m_j &\le \frac{1}{n}\sum_{i=1}^{n} Z_i^{(j)} + \sqrt{\frac{2B^2 m_j u}{n}} + \frac{2B^2 u}{3n} \\ &\le \frac{1}{n}\sum_{i=1}^{n} Z_i^{(j)} + \frac{m_j}{2} + \frac{5B^2 u}{3n}, \end{aligned}$$

where the second line uses $\sqrt{2ma} \le m/2 + a$. Consequently,

$$m_j \le 2\widehat{\mathcal{L}}_{\mathrm{traj}}(\bar{\theta}_j) + \frac{10B^2 u}{3n}. \tag{15}$$

For any θ, choose $\bar{\theta} \in \Theta_\alpha$ with $\|\theta - \bar{\theta}\| \le \alpha$. By Eqs. (13)–(14),

$$|f_\theta(\tau)^2 - f_{\bar{\theta}}(\tau)^2| \le |f_\theta(\tau) - f_{\bar{\theta}}(\tau)|\,|f_\theta(\tau) + f_{\bar{\theta}}(\tau)| \le 2BL\alpha. \tag{16}$$

Applying this bound to both population and empirical losses yields

$$\begin{aligned} \mathcal{L}_{\mathrm{traj}}(Q_\theta) &\le \mathcal{L}_{\mathrm{traj}}(Q_{\bar{\theta}}) + 2BL\alpha \\ &\le 2\widehat{\mathcal{L}}_{\mathrm{traj}}(\bar{\theta}) + \frac{10B^2 u}{3n} + 2BL\alpha \\ &\le 2\widehat{\mathcal{L}}_{\mathrm{traj}}(\theta) + \frac{10B^2 u}{3n} + 6BL\alpha. \end{aligned}$$

This holds simultaneously for every θ and proves the claim. The $1/n$ statistical term comes from the variance–mean relation above, rather than an absolute Hoeffding bound; realizability is used only later to obtain a zero-loss comparator. □

Proof of Theorem 3. By Corollary 1, applied with $Q = \widehat{Q}$,

$$J(\pi^\star) - J(\pi_{\widehat{Q}}) \le 2 \max_{\pi \in \{\pi^\star, \pi_{\widehat{Q}}\}} \left| \mathbb{E}_{\tau\sim\mathbb{P}^\pi}\left[G_{\widehat{Q},\xi^{\widehat{Q}}}(\tau) - R(\tau)\right] \right|. \tag{17}$$

Next, by Corollary 2, for every $\pi \in \{\pi^\star, \pi_{\widehat{Q}}\}$,

$$\left(\mathbb{E}_{\tau\sim\mathbb{P}^\pi}\left[G_{\widehat{Q},\xi^{\widehat{Q}}}(\tau) - R(\tau)\right]\right)^2 \le \left(1 + \sum_{h=0}^{H-1} \chi^2(d_h^\pi \| d_h^{\pi_b})\right) \mathcal{L}_{\mathrm{traj}}(\widehat{Q}).$$

By the definition of the concentrability coefficient,

$$C_{\mathrm{conc}}(\pi_b) := 1 + \sup_{Q\in\mathcal{Q}} \sup_{\pi\in\{\pi^\star,\pi_Q\}} \sum_{h=0}^{H-1} \chi^2(d_h^\pi \| d_h^{\pi_b}),$$

we therefore have

$$\left(\mathbb{E}_{\tau\sim\mathbb{P}^\pi}\left[G_{\widehat{Q},\xi^{\widehat{Q}}}(\tau) - R(\tau)\right]\right)^2 \le C_{\mathrm{conc}}(\pi_b)\,\mathcal{L}_{\mathrm{traj}}(\widehat{Q}) \qquad \forall \pi \in \{\pi^\star, \pi_{\widehat{Q}}\}.$$

Taking square roots and substituting into (17) yields

$$J(\pi^\star) - J(\pi_{\widehat{Q}}) \le 2\sqrt{C_{\mathrm{conc}}(\pi_b)\,\mathcal{L}_{\mathrm{traj}}(\widehat{Q})}. \tag{18}$$

It remains to upper bound $\mathcal{L}_{\mathrm{traj}}(\widehat{Q})$. By Lemma 7, with probability at least $1-\delta$, uniformly over all $Q \in \mathcal{Q}$,

$$\mathcal{L}_{\mathrm{traj}}(Q) \le 2\widehat{\mathcal{L}}_{\mathrm{traj}}(Q) + \varepsilon_{\mathrm{stat}}(n, \alpha, \delta).$$

Applying this bound to $Q = \widehat{Q}$, we obtain

$$\mathcal{L}_{\mathrm{traj}}(\widehat{Q}) \le 2\widehat{\mathcal{L}}_{\mathrm{traj}}(\widehat{Q}) + \varepsilon_{\mathrm{stat}}(n, \alpha, \delta). \tag{19}$$

By the approximate ERM assumption in Theorem 3,

$$\widehat{\mathcal{L}}_{\mathrm{traj}}(\widehat{Q}) \le \inf_{Q\in\mathcal{Q}} \widehat{\mathcal{L}}_{\mathrm{traj}}(Q) + \varepsilon_{\mathrm{opt}}.$$

Therefore

$$\mathcal{L}_{\mathrm{traj}}(\widehat{Q}) \le 2 \inf_{Q\in\mathcal{Q}} \widehat{\mathcal{L}}_{\mathrm{traj}}(Q) + 2\varepsilon_{\mathrm{opt}} + \varepsilon_{\mathrm{stat}}(n,\alpha,\delta). \tag{20}$$

Now, by Assumption 1, we have $Q^\star \in \mathcal{Q}$. Soft Bellman optimality in (2) and the definition of the auxiliary best response give

$$Q^\star(s,a) - \gamma\, \xi^{Q^\star}(s,a) = Q^\star(s,a) - \gamma\, \mathbb{E}[V^\star(s') \mid s,a] = R(s,a).$$

Consequently,

$$G_{Q^\star,\xi^{Q^\star}}(\tau) = \sum_{h=0}^{H-1} \gamma^h R(s_h,a_h) = R(\tau) \qquad \text{for } \mathbb{P}^{\pi_b}\text{-a.e. } \tau.$$

Therefore,

$$\widehat{\mathcal{L}}_{\mathrm{traj}}(Q^\star) = 0.$$

This argument only requires $Q^\star$ to be a feasible zero-loss comparator. Since $Q^\star \in \mathcal{Q}$, this implies

$$\inf_{Q\in\mathcal{Q}} \widehat{\mathcal{L}}_{\mathrm{traj}}(Q) \le \widehat{\mathcal{L}}_{\mathrm{traj}}(Q^\star) = 0.$$

Substituting this into (20), we conclude that

$$\mathcal{L}_{\mathrm{traj}}(\widehat{Q}) \le 2\varepsilon_{\mathrm{opt}} + \varepsilon_{\mathrm{stat}}(n,\alpha,\delta). \tag{21}$$

Finally, plugging (21) into (18) gives

$$J(\pi^\star) - J(\pi_{\widehat{Q}}) \le 2\sqrt{C_{\mathrm{conc}}(\pi_b)\,(2\varepsilon_{\mathrm{opt}} + \varepsilon_{\mathrm{stat}}(n,\alpha,\delta))},$$

which proves the first claim.

For the simplified rate, under the standard entropy growth condition

$$\log \mathcal{N}(\Theta,\alpha) \lesssim \dim(\Theta)\log(C/\alpha),$$

choosing $\alpha = n^{-1}$ in Lemma 7 gives

$$\varepsilon_{\mathrm{stat}}(n,n^{-1},\delta) \lesssim \frac{B^2\big(\dim(\Theta)\log(Cn) + \log(2/\delta)\big) + BL}{n} = \widetilde{O}\left(\frac{\dim(\Theta)}{n}\right),$$

for fixed B, L and $\dim(\Theta) \ge 1$. Hence

$$J(\pi^\star) - J(\pi_{\widehat{Q}}) \le \widetilde{O}\left(\sqrt{C_{\mathrm{conc}}(\pi_b)\left(\varepsilon_{\mathrm{opt}} + \frac{\dim(\Theta)}{n}\right)}\right).$$

□

A.3 Optimization guarantee

As in Section 4.2, the dataset and initialization are fixed. $\widehat{\mathbb{E}}_{\mathcal{D}}$ averages stored trajectories, whereas $\mathbb{E}$ averages optimization randomness. The following decomposition is used to prove Theorem 4.

Proposition 1 (Bias decomposition of the double stop-gradient field)**.** *Under Assumption 4, for every pair* (θ,ϕ)*,*

$$\|\nabla_\theta F(\theta) - g_{\mathrm{DSG}}(\theta,\phi)\|_2 \;\le\; 2\gamma C_H G_Q \sqrt{F(\theta)} + 2\gamma C_H^2 G_Q \|\xi_\phi - \xi^\theta\|_\infty. \tag{22}$$

Consequently,

$$\|\nabla_\theta F(\theta) - g_{\mathrm{DSG}}(\theta,\phi)\|_2^2 \;\le\; 8\gamma^2 C_H^2 G_Q^2\, F(\theta) + 8\gamma^2 C_H^4 G_Q^2\, \|\xi_\phi - \xi^\theta\|_\infty^2. \tag{23}$$

Proof of Proposition 1. Define the exact-target frozen field

$$g_\star(\theta) := 2\widehat{\mathbb{E}}_{\mathcal{D}}\left[\left(G_{Q_\theta,\xi^\theta}(\tau) - R(\tau)\right)\sum_{h=0}^{H-1}\gamma^h \nabla_\theta Q_\theta(s_h,a_h)\right].$$

This names a field; it does not assume an auxiliary parameter realizes ξ^θ. For shorthand, write

$$\epsilon_\theta(\tau) := G_{Q_\theta,\xi^\theta}(\tau) - R(\tau), \qquad \epsilon_{\theta,\phi}(\tau) := G_{Q_\theta,\xi_\phi}(\tau) - R(\tau).$$

By the chain rule,

$$\nabla_\theta F(\theta) = 2\,\widehat{\mathbb{E}}_{\mathcal{D}}\left[\epsilon_\theta(\tau)\,\nabla_\theta G_{Q_\theta,\xi^\theta}(\tau)\right].$$

Moreover,

$$\nabla_\theta G_{Q_\theta,\xi^\theta}(\tau) = \sum_{h=0}^{H-1}\gamma^h\Big(\nabla_\theta Q_\theta(s_h,a_h) - \gamma\,\nabla_\theta \xi^\theta(s_h,a_h)\Big).$$

Hence

$$\nabla_\theta F(\theta) - g_\star(\theta) = -2\gamma\,\widehat{\mathbb{E}}_{\mathcal{D}}\left[\epsilon_\theta(\tau)\sum_{h=0}^{H-1}\gamma^h\nabla_\theta\xi^\theta(s_h,a_h)\right]. \tag{24}$$

Next, by definition of ξ^θ,

$$\nabla_\theta\xi^\theta(s,a) = \mathbb{E}_{s'\sim P(\cdot|s,a)}[\nabla_\theta V_{Q_\theta}(s')].$$

Also,

$$\nabla_\theta V_{Q_\theta}(s) = \sum_{a\in\mathcal{A}}\pi_{Q_\theta}(a\mid s)\nabla_\theta Q_\theta(s,a),$$

at nonterminal states, with zero derivative at termination. Thus Assumption 4 implies

$$\|\nabla_\theta\xi^\theta(s,a)\|_2 \le G_Q.$$

Using (24), Cauchy–Schwarz, and $\sum_{h=0}^{H-1}\gamma^h = C_H$, we obtain

$$\begin{aligned}\|\nabla_\theta F(\theta) - g_\star(\theta)\|_2 &\le 2\gamma C_H G_Q\,\widehat{\mathbb{E}}_{\mathcal{D}}[|\epsilon_\theta(\tau)|]\\ &\le 2\gamma C_H G_Q\sqrt{\widehat{\mathbb{E}}_{\mathcal{D}}[\epsilon_\theta(\tau)^2]} = 2\gamma C_H G_Q\sqrt{F(\theta)}.\end{aligned} \tag{25}$$

It remains to control the tracking mismatch $\xi_\phi - \xi^\theta$. By (4.2),

$$g_{\mathrm{DSG}}(\theta,\phi) - g_\star(\theta) = 2\,\widehat{\mathbb{E}}_{\mathcal{D}}\left[\left(\epsilon_{\theta,\phi}(\tau) - \epsilon_\theta(\tau)\right)\sum_{h=0}^{H-1}\gamma^h\nabla_\theta Q_\theta(s_h,a_h)\right].$$

Since

$$\epsilon_{\theta,\phi}(\tau) - \epsilon_\theta(\tau) = -\gamma\sum_{h=0}^{H-1}\gamma^h\big(\xi_\phi - \xi^\theta\big)(s_h,a_h),$$

we have

$$|\epsilon_{\theta,\phi}(\tau) - \epsilon_\theta(\tau)| \le \gamma C_H\|\xi_\phi - \xi^\theta\|_\infty.$$

Together with

$$\left\|\sum_{h=0}^{H-1}\gamma^h\nabla_\theta Q_\theta(s_h,a_h)\right\|_2 \le C_H G_Q,$$

this yields

$$\|g_{\mathrm{DSG}}(\theta,\phi) - g_\star(\theta)\|_2 \le 2\gamma C_H^2 G_Q\|\xi_\phi - \xi^\theta\|_\infty. \tag{26}$$

Combining (25) and (26) by the triangle inequality proves (22). The squared bound (23) follows from $(x+y)^2 \le 2x^2 + 2y^2$. □

Proof of Theorem 4. Let

$$e_t := g_{\mathrm{DSG}}(\theta_t, \phi_t) - \nabla_\theta F(\theta_t).$$

By Assumption 3 and standard smoothness of F,

$$F(\theta_{t+1}) \le F(\theta_t) - \eta \langle \nabla_\theta F(\theta_t), \hat{g}_t \rangle + \frac{L_F \eta^2}{2} \|\hat{g}_t\|_2^2.$$

Taking conditional expectation with respect to $\mathcal{F}_t$, using Assumption 5, and noting that

$$\mathbb{E}[\hat{g}_t \mid \mathcal{F}_t] = \nabla_\theta F(\theta_t) + e_t,$$

gives

$$\begin{aligned}\mathbb{E}[F(\theta_{t+1}) \mid \mathcal{F}_t] \le F(\theta_t) &- \eta \left\langle \nabla_\theta F(\theta_t), \nabla_\theta F(\theta_t) + e_t \right\rangle \\ &+ \frac{L_F \eta^2}{2} \mathbb{E}\big[\|\hat{g}_t\|_2^2 \mid \mathcal{F}_t\big]. \end{aligned} \tag{27}$$

Also,

$$\mathbb{E}\big[\|\hat{g}_t\|_2^2 \mid \mathcal{F}_t\big] = \|\nabla_\theta F(\theta_t) + e_t\|_2^2 + \mathbb{E}\big[\|\hat{g}_t - g_{\mathrm{DSG}}(\theta_t, \phi_t)\|_2^2 \mid \mathcal{F}_t\big] \le 2\|\nabla_\theta F(\theta_t)\|_2^2 + 2\|e_t\|_2^2 + \sigma^2,$$

and

$$\langle \nabla_\theta F(\theta_t), \nabla_\theta F(\theta_t) + e_t \rangle \ge \frac{1}{2}\|\nabla_\theta F(\theta_t)\|_2^2 - \frac{1}{2}\|e_t\|_2^2.$$

Substituting these bounds into (27) yields

$$\mathbb{E}[F(\theta_{t+1}) \mid \mathcal{F}_t] \le F(\theta_t) - \Big(\frac{\eta}{2} - L_F \eta^2\Big)\|\nabla_\theta F(\theta_t)\|_2^2 + \Big(\frac{\eta}{2} + L_F \eta^2\Big)\|e_t\|_2^2 + \frac{L_F \eta^2}{2}\sigma^2.$$

Since $\eta \le 1/(4L_F)$,

$$\frac{\eta}{2} - L_F \eta^2 \ge \frac{\eta}{4}, \qquad \frac{\eta}{2} + L_F \eta^2 \le \frac{3\eta}{4}.$$

Taking full expectation, summing from $t = 0$ to $T - 1$, and telescoping gives

$$\frac{\eta}{4} \sum_{t=0}^{T-1} \mathbb{E}\big[\|\nabla_\theta F(\theta_t)\|_2^2\big] \le F(\theta_0) - F_{\inf} + \frac{3\eta}{4} \sum_{t=0}^{T-1} \mathbb{E}[\|e_t\|_2^2] + \frac{L_F \eta^2 \sigma^2}{2} T.$$

Dividing by $\eta T/4$ and using

$$\min_{0 \le t \le T-1} \mathbb{E}\big[\|\nabla_\theta F(\theta_t)\|_2^2\big] \le \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\big[\|\nabla_\theta F(\theta_t)\|_2^2\big],$$

we obtain

$$\min_{0 \le t \le T-1} \mathbb{E}\big[\|\nabla_\theta F(\theta_t)\|_2^2\big] \le \frac{4\big(F(\theta_0) - F_{\inf}\big)}{\eta T} + 2L_F \eta \sigma^2 + \frac{3}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|e_t\|_2^2]. \tag{28}$$

Finally, Proposition 1 implies

$$\|e_t\|_2^2 \le 8\gamma^2 C_H^2 G_Q^2 \, F(\theta_t) + 8\gamma^2 C_H^4 G_Q^2 \, \|\xi_{\phi_t} - \xi^{\theta_t}\|_\infty^2.$$

Substituting this into (28) and using (11) yields (12). □

B EXPERIMENTAL DETAILS

B.1 CONTROLLED MDPS

FrozenLake uses `FrozenLake-v1` with the 8x8 map and `is_slippery=True`. The dataset is a 30:70 mixture of 20,000 trajectories: 30% are generated by the hard value-iteration greedy policy and 70% by an ϵ-greedy variant with $\epsilon = 0.2$ that explores over non-greedy actions. The soft-VI reference uses $\gamma = 0.99$ and entropy parameter $\beta = 1$. Stochastic-TBRM receives the trajectory return label and the transition sequence only; it is not given per-transition rewards. The tabular Q and ξ tables are trained with 20 auxiliary updates per critic update. FQI, CQL, Off-GLADIUS, and OptiDICE are run in their standard offline-RL reward interface. All methods are trained for 50,000 iterations with evaluation every 5,000 iterations and 10 random seeds.

Acrobot uses `Acrobot-v1` with a fixed 30:70 offline replay dataset whose mean episode return is -375.71. Stochastic-TBRM uses neural Q and ξ networks, receives only the aggregate trajectory return for each episode, and uses the same 20:1 auxiliary-to-critic update schedule. The reported Acrobot curves and tables use 50,000 training iterations, evaluation every 5,000 iterations, 20 evaluation episodes per checkpoint, and 10 random seeds.

Auxiliary update schedule. Figure 2 varies the number of ξ updates per critic update on FrozenLake. All three schedules reach the same final Q-MAE, but one update per iteration converges roughly three times more slowly than ten or twenty, which are indistinguishable. This matches the role of ξ in the objective. The critic gradient is unbiased only when ξ tracks $\mathbb{E}[V(s') \mid s, a]$, and a single inner step leaves ξ lagging behind the current V. The main experiments use twenty updates.

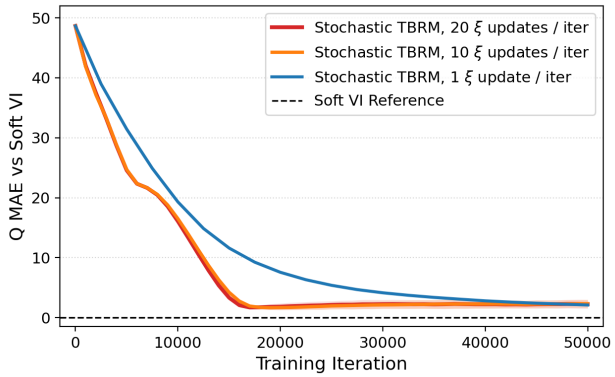

Figure 2: FrozenLake Q-MAE against exact soft value iteration for one, ten, and twenty ξ updates per critic update. Mean over 10 seeds.

B.2 AGENT DATA AND ENVIRONMENTS

Sources, splits, and information boundary. We use the public DMPO implementation of WebShop, ScienceWorld, and ALFWorld. The reported evaluations contain 200 WebShop test episodes, all 211 ScienceWorld tasks in the released test-index file, and all 134 ALFWorld test tasks. ScienceWorld and ALFWorld have 194 and 140 development tasks, respectively. All three methods are trained from the same released expert demonstrations on the training split. DMPO and Q-Evolve additionally collect their own rollouts, in both cases on training tasks only. We used training data to fit models and development data to choose checkpoints and interface variants; test tasks were evaluated only after those choices were frozen. All state features and executable candidates are functions of the task text, action history, and observations exposed to an ordinary agent. They do not use hidden simulator fields, answer keys, evaluator state, or task-specific test scripts. The same public prompt interface is the only input to the baseline policies.

Macro-action spaces. The critic has a fixed output head over the union of benchmark actions and masks outputs to the active benchmark. A macro action constrains a family of public commands. It is

not itself sent to the simulator. On WebShop and ALFWorld the critic is the primary controller and selects

$$a_h \in \arg \max_{a \in \mathcal{A}_{\text{valid}}(s_h)} Q(s_h, a)$$

where $\mathcal{A}_{\text{valid}}(s_h)$ is the set of macro actions realizable from the current public observation. ScienceWorld uses the same decision only as the conservative shield after an exact-command proposal is not executable. Table 3 records the exact action families. The baselines do not use this head. They generate commands directly from the public prompt.

Table 3: Benchmark-specific macro actions and their command-level interpretation.

Benchmark	Macro actions and public command families
WebShop	`search`, `click_product`, `click_option`, `click_navigation`, `click_info`, and buy.
ScienceWorld	`sw_explore` (look around, teleport, open/close), `sw_target` (focus, look at, examine, read), `sw_collect` (pick up/take), `sw_place` (put/move/drop), and `sw_operate` (remaining interactions such as activate, connect, use, pour, or mix).
ALFWorld	`aw_search` (navigate/open and inspect), `aw_retrieve` (take the target), `aw_transform` (clean, heat, cool, or toggle), `aw_place` (put the target), and `aw_recover` (inventory, observation, and recovery actions).

State construction. For each assistant turn, s_h is the public prefix ending at the previous environment observation and a_h is the macro label parsed from the next assistant command. WebShop adds page type, buy availability, visible product identifiers, selected-product status, and selected-option memory. ScienceWorld adds target, container, held-object, and recent-action summaries recovered from public text. ALFWorld similarly records known receptacles, current location, held/deposited targets, and required transformations. A pending ALFWorld take or put is promoted to confirmed state only after the subsequent observation supports it. Opening state and lamp location/power are handled in the same observation-confirmed manner. These rules are benchmark-level parsers shared across tasks rather than task-specific plans.

Replay construction. The source demonstrations are converted to transition sequences but retain only their scalar terminal label as reward supervision. The WebShop replay uses expert demonstrations only. The ScienceWorld replay attached to the evaluated checkpoint combines 1,429 audited expert trajectories with 384 train-only rollouts, and the ALFWorld replay combines 3,119 audited expert trajectories with 128 train-only rollouts. Table 4 reports the resulting replay for each evaluated checkpoint. BC warm starts use the full-score ScienceWorld subset of 1,434 trajectories and the successful ALFWorld subset of 3,152 trajectories. Lower-outcome trajectories remain available to the subsequent outcome-balanced Stochastic-TBRM stage.

Table 4: Audited replay associated with each evaluated critic. The final column uses the benchmark-valid high-outcome partition: full normalized score for ScienceWorld and the win indicator for ALFWorld.

Benchmark	Trajectories	Transitions	Avg. length	Avg. label	High/other
WebShop	1,824	7,103	3.89	0.830	–
ScienceWorld	1,813	26,004	14.34	0.789	1,434/379
ALFWorld	3,247	35,160	10.83	0.971	3,152/95

Leakage audit. The replay builder subtracts the authoritative development and test identifiers, rejects unknown identifiers, records source hashes, and verifies the benchmark action schema. It found no held-out overlap in the ScienceWorld or ALFWorld replay. The released WebShop training

index shares 21 identifiers with its 200-episode test index, so the expert demonstrations inherited identically by all three methods contain those tasks. We therefore report WebShop as a locked evaluation rather than held-out evidence. Q-Evolve collects rollouts on the released training split only, with no overlap against any test set.

B.3 Training and selection

ScienceWorld exact-command policy. We first fine-tune a rank-16 Qwen3.5-4B LoRA (scaling 32, zero dropout) on 19,240 command examples from the 1,434 full-score train trajectories. The targets are exact public commands; the input renderer removes `[TaskId]` before tokenization. This sanitized stage uses one epoch (301 optimizer updates), per-device batch size 8, gradient accumulation 8, learning rate 2×10^{-5}, Adafactor with a cosine schedule and nine warmup steps, bf16, gradient checkpointing, maximum length 2,048, and seed 42. The final adapter continues this LoRA for one epoch at learning rate 5×10^{-6} on 10,800 examples: 1,200 examples from each of nine generic public goal-signature strata. It retains effective batch size 64, uses five warmup steps, and covers all 1,429 expert train task identifiers. LoRA targets `q/k/v/o/gate/up/down_proj`. A three-episode train-only plumbing diagnostic fixed the deployed list-assisted prompt before online development evaluation; no development or test command enters either SFT stage.

Critic architecture and optimization. The Q and ξ functions share the bf16 `Skywork/Skywork-Reward-V2-Llama-3.1-8B` backbone at revision `cba2f842f3f1af2f1b2f0d35e794d789976390c5`. We train rank-16 LoRA adapters with scaling factor 32, zero dropout, and target modules `q_proj`, `k_proj`, `v_proj`, and `o_proj`; Q and ξ use separate linear heads. The configured Stochastic-TBRM stage uses Adafactor for Q, AdamW for ξ, zero weight decay, $\gamma = 0.8$, four auxiliary updates per outer iteration, trajectory batch size 8, transition batch size 64, maximum state length 2,048, and seed 42. For the five ScienceWorld iterations, 25% of each balanced trajectory batch is sampled from the replay's retained `environment_done=false` stratum. This Boolean affects sampling only; every regression target remains the scalar normalized shaped outcome.

Table 5: Optimization and selection for the evaluated interactive-agent checkpoints. "Selected iter." counts post-BC Stochastic-TBRM iterations in the checkpoint used on test.

Benchmark	BC steps	Selected iter.	LR(Q)	LR(ξ)	Replay cap.
WebShop	200	100	2×10^{-5}	5×10^{-5}	512
ScienceWorld	100	5	1×10^{-5}	1×10^{-4}	512
ALFWorld	100	0	5×10^{-5}	1×10^{-4}	512

Checkpoint selection. Behavior cloning fits the macro-action head by masked softmax cross entropy before Stochastic-TBRM optimization. We compared saved checkpoints only on development data and fixed the selected checkpoint before test evaluation. WebShop selected iteration 100 after its warm start. ScienceWorld selected iteration 5 together with the train-only balanced exact-command adapter; the critic backbone-adapter and head SHA-256 prefixes are `ad67aa9bcbe4` and `0537962d333b`, and the final command-policy weights prefix is `b0e58d4cf7b7`. The otherwise identical BC-shield screen also beat DMPO, so the deployed pipeline does not isolate a gain caused specifically by the five post-BC updates. ALFWorld retains its 100-step BC warm start (adapter prefix `78ffd54d65e9`) because later Stochastic-TBRM checkpoints did not improve its locked development criterion. Baseline checkpoints were not selected on development data. DMPO uses its final training epoch, and Q-Evolve uses the loop count prescribed by its authors (Appendix B.4).

ScienceWorld interface ablations. Table 6 reports the complete 36-episode screening partition (development indices 12–47) under the same full-inventory, 128-command-shortlist contract. A direct transfer of ALFWorld's phase-conditioned compositional executor fails because ScienceWorld's broad `operate` phase admits many syntactically valid but semantically incompatible object–relation combinations. Preserving exact-command generation and using the critic conservatively is the material interface change. Both shield checkpoints beat DMPO; the post-TBRM checkpoint is retained for the later balanced-policy study because it has the larger paired reward gain.

Table 6: ScienceWorld interface screening. Differences and confidence intervals are paired against the matched DMPO reference (reward 0.3341) on the same 36 development episodes.

Variant	Reward	Difference	95% CI
ALF-style dynamic executor (BC)	0.2526	-0.0815	$[-0.1872, 0.0209]$
ALF-style dynamic executor (post-TBRM)	0.2366	-0.0976	$[-0.2127, 0.0139]$
Exact-command policy, no shield	0.3480	$+0.0139$	$[-0.1663, 0.1886]$
Exact-command policy + BC shield	0.5325	$+0.1984$	$[+0.0719, +0.3183]$
Exact-command policy + post-TBRM shield	**0.5442**	$+\mathbf{0.2101}$	$[+0.0714, +0.3403]$

After the initial candidate failed its untouched confirmation (Appendix B.4), all later development experiments are exploratory. Table 7 reports the evidence used to freeze the final interface. Balanced exact-command training plus a scoped manipulation-repeat guard improves reward on both inspected partitions. In contrast, transferring a larger ALFWorld-inspired guard stack perturbs the chemistry shortlist and loses full-score completions, so those guards are disabled in the final method. On indices 146–193, the full guard stack reduces full-score completion by 6.25 percentage points relative to DMPO despite a small reward difference of $+0.0067$ whose confidence interval includes zero.

Table 7: Final ScienceWorld exploratory ablations. Each difference is paired against matched DMPO on the indicated development indices. The selected candidate enables only the scoped repeat guard.

Variant	Indices	Reward	Difference	95% CI
Balanced policy + scoped repeat	12–47	0.5304	$+0.1963$	$[+0.0800, +0.3040]$
Balanced policy + scoped repeat	146–193	0.4237	$+0.0929$	$[+0.0028, +0.1834]$
Balanced policy + full public guards	146–193	0.3375	$+0.0067$	$[-0.0890, +0.1014]$

B.4 Evaluation

Matched generator contract. ScienceWorld and ALFWorld use `Qwen/Qwen3.5-4B` revision `851bf6e806efd8d0a36b00ddf55e13ccb7b8cd0a` in bf16 with temperature 0, top-$p = 1$, no sampling, and at most 192 new tokens. Our method and DMPO additionally share a single serving stack. On ScienceWorld they use maximum context length 8,192, batch size one, and no collection window. The candidate uses its train-only balanced exact-command adapter (served-identity prefix `156487a8d042`), DMPO uses its preference-trained adapter (prefix `73926965bcf1`), and Q-Evolve uses its final-loop adapter (prefix `866744e2f888`). For our method and DMPO, the ALFWorld contract remains maximum context length 32,768, batch size four, and a 20-ms window. Our ALFWorld method uses the base generator, DMPO uses adapter prefix `7160ff44a21e`, and Q-Evolve uses adapter prefix `db72ab19c863`. Because Q-Evolve merges its LoRA weights at the end of each loop, its adapters sit on a merged base rather than on the released revision. A strict paired validator checks the ordered tasks, split, seed, generator revision, decoding settings, action schema, public interface, shortlist semantics, simulator JAR, and task-family step caps before computing differences. This validator applies to the comparison between our method and DMPO. The Q-Evolve comparison is matched on a smaller set of attributes, described below.

Macro-to-micro execution. In ScienceWorld, our exact-command policy receives the public state and a progress-ranked shortlist of 128 commands, while its output is normalized and checked against the complete public executable inventory. A valid proposal is preserved. Otherwise, the post-TBRM critic selects one of the five phases and the generator chooses among up to 128 compatible public commands. Numeric replies require an observed ambiguity menu; redundant current-room navigation is removed; and a publicly failed/no-progress command is cooled down for one turn. The final interface additionally blocks an immediate repeated action only in manipulation task families. The no-progress cycle, typed-action, inventory-qualifier, goal-focus, and candidate-conditioned-Q guards are disabled. In ALFWorld, a deterministic executor constructs at most eight valid commands for the selected phase from the task text, current observation, and observation-confirmed memory. WebShop uses the generic Qwen micro policy and disables action repair, learned click rankers, structured product scoring, and structured option scoring.

DMPO-Qwen training. For ScienceWorld and ALFWorld, we train the Qwen SFT adapter for three epochs on the released training split, collect SFT-policy rollouts on training tasks, pair them with expert trajectories, and train DMPO for three epochs. DMPO uses effective batch size 32, learning rate 10^{-6}, $\beta = 0.1$, and $\gamma = 0.7$, producing 1,444 ScienceWorld and 2,173 ALFWorld preference pairs. The WebShop baseline uses 153 pairs from 200 Qwen rollouts and 200 optimizer steps. All reported DMPO policies use the matched greedy generator contract described above. The selected ScienceWorld DMPO adapter and served identity are hash-pinned. Its retained training summary records the 1,444 pairs and optimization hyperparameters, but the named raw preference JSONL is no longer present; evaluation from the released weights is reproducible, whereas independent reconstruction of that adapter from its preference examples is not.

Q-Evolve training. We use the official implementation with the released expert demonstrations and the same Qwen3.5-4B starting checkpoint, following the loop counts prescribed by the authors: three for WebShop and two for ScienceWorld and ALFWorld. Each loop collects rollouts with the current policy on training tasks, fits an IQL-style critic on the expert and rollout data, and updates the policy with step-wise advantages under a clipped behavior-proximal objective. Q-Evolve uses its own ReAct prompting with one in-context example and passes generated text directly to the simulator, so it shares neither our macro-action schema nor the 128-command shortlist. Its comparison is matched on the ordered task list, split, seed 42, greedy decoding, the 192-token limit, and step caps rather than on the full paired-validator contract. Table 8 reports every loop, and Table 2 reports the prescribed loop in all three cases.

Table 8: Q-Evolve across the self-evolution loop counts prescribed by its authors (three for WebShop, two for ScienceWorld and ALFWorld). Loop 0 is the behavior-cloning start. Bold marks the prescribed loop, which Table 2 reports. ALFWorld is non-monotone across loops. "Completion" is full normalized score on ScienceWorld and task success otherwise.

Benchmark	Loop	Reward	Completion	Steps
WebShop	0 (BC)	0.581	0.830	4.75
	1	0.590	0.845	4.65
	2	0.643	0.900	4.73
	3	**0.674**	**0.940**	4.51
ScienceWorld	0 (BC)	0.503	0.213	20.43
	1	0.514	0.251	21.67
	2	**0.520**	**0.265**	21.95
ALFWorld	0 (BC)	0.493	0.493	26.04
	1	0.567	0.567	23.27
	2	**0.463**	**0.463**	25.71

Paired statistics. All intervals in this appendix are 95% percentile intervals from 10,000 task-level bootstrap resamples drawn with replacement using seed 1729, computed on per-task differences against the matched DMPO reference. The ScienceWorld wrapper's legacy `success` field is an environment-done indicator that can also be set after a negative score, and is reported only as a diagnostic.

ScienceWorld development audit. We predeclared development indices 12–47 for screening, 0–145 for broad selection, and 146–193 as an untouched confirmation partition. The reward-oriented pipeline was selected on indices 0–145 and evaluated once on indices 146–193. The confirmation difference was $+0.0321$ with 95% interval $[-0.0585, +0.1196]$, below the predeclared +0.10 gate and with a nonpositive lower bound. The automated controller therefore did not open the 211-task test for that initial candidate. A subsequent read-only audit found a large public task-family shift and a 71.2% immediate-repeat rate for our policy; 1,059 of 1,065 repeats had been accepted as valid exact-command proposals. All later ScienceWorld development experiments are labeled exploratory, and no second "untouched confirmation" claim is made. The final balanced-policy scoped-repeat candidate improves reward on both inspected development subsets: $+0.1963$ with interval $[+0.0800, +0.3040]$ on indices 12–47 and $+0.0929$ with interval $[+0.0028, +0.1834]$ on indices 146–193. The final

configuration was frozen from this development evidence before its 211-task comparison. No partial test outcome was used for selection or scheduling. Deterministic sharding used episode indices only.

Reproducibility identifiers. The full hashes and run manifests accompany the released evaluation artifacts. The final ScienceWorld episode hashes are `90e11d765248` (ours), `e24c08e79d56` (DMPO), and `db78234c3d48` (Q-Evolve), and its pre-test freeze manifest hash is `3d42a39275df`. The held-out ALFWorld episode hashes are `ea80c7958667` (ours), `74a84725d648` (DMPO), and `b9398e54dc8d` (Q-Evolve). The WebShop Q-Evolve episode hash is `273b6a1e1323`, and the three Q-Evolve summary hashes are `7f0571f8c904`, `34ee7a45491c`, and `69acb53744f9`. Every summary records the exact means, ordered task IDs, generator metadata, adapter identity, and evaluation contract, and each Q-Evolve summary additionally records the merged base underlying its adapter. ScienceWorld's deterministic shards are merged only after validation of contiguous indices, common checkpoint, and common endpoint identity, and the final summary retains per-shard provenance.

Offline diagnostics. During development, we also reloaded candidate critics and measured induced-return prediction error on sampled replay trajectories. This was a checkpoint sanity check only: it did not query a simulator and is not reported as a benchmark result.

C Related work

Multi-turn agentic alignment. Recent advancements in large language model (LLM) alignment have fundamentally shifted the paradigm from myopic, single-turn generation to complex, long-horizon sequential decision-making. Traditional frameworks, such as Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Ouyang et al., 2022) and Direct Preference Optimization (DPO) (Rafailov et al., 2023), were initially designed to optimize isolated prompt-response pairs. However, as LLMs are increasingly deployed as interactive agents capable of tool use and extended reasoning, they must maintain alignment over multi-turn trajectories characterized by delayed and sparse feedback (Glaese et al., 2022). To bridge this gap, recent studies have extended preference optimization to multi-turn settings to account for trajectory-level normalization (Shani et al., 2024; Shi et al., 2024), while complementary works have explored explicit process-level credit assignment to mitigate variance in long-horizon tasks (Zeng et al., 2025; Peng et al., 2026). A related line iteratively improves the policy by collecting fresh rollouts and refitting a critic on them, which converts outcome labels into step-wise advantages at the cost of repeated environment access (Zhang et al., 2026). Despite this rapid progress, acquiring dense, step-by-step human annotation for every agentic action is cost-prohibitive. Consequently, practical supervision is frequently limited to episode-level outcome scores, making accurate step-level credit assignment a central bottleneck that requires more sample-efficient, offline approaches.

Offline reinforcement learning. Offline reinforcement learning (RL) provides a mathematical framework for learning optimal policies strictly from static, pre-collected datasets, avoiding the need for computationally expensive online interactions (Levine et al., 2020). However, offline RL is unstable due to the "deadly triad" (Sutton et al., 1998), which combines off-policy learning, function approximation, and bootstrapping. This combination exacerbates distribution shifts and causes severe extrapolation errors. To mitigate this, seminal works in traditional RL introduced behavior-regularization to safely extract robust policies from suboptimal logs (Fujimoto et al., 2019). Recently, this exact paradigm has gained significant traction for aligning large language models (LLMs), often in the form of KL-divergence constraints against a reference model. Because traditional online RLHF requires continuous rollouts and repetitive reward querying, framing alignment as an offline decision-making process allows models to be optimized directly from static, pre-existing trajectories. This offers a highly scalable and sample-efficient alternative (Baheti et al., 2023; Ji et al., 2024), though effectively applying these principles to the sequential nature of language generation requires specialized objective functions.

Trajectory Bellman residual minimization. A notable development in value-based offline alignment is the adaptation of Bellman Residual Minimization (BRM) to the trajectory level. Minimizing the temporal difference error to enforce Bellman consistency has long been established in classical reinforcement learning with function approximation (Baird, 1995; Antos et al., 2008). To address the high sample complexity of online alignment, several studies have sought to bridge these classical value-based methods with large language models by deriving loss functions directly from the Bellman equation (Clavier et al., 2025). Building on this foundation, later work lifted the step-wise BRM concept to a sequence-level objective. This advancement, known as Trajectory Bellman Residual Minimization (TBRM), bypasses the prohibitive cost of dense reward labeling, enabling simple and highly efficient optimization directly from episodic, outcome-annotated data (Yuan et al., 2025). By enforcing soft Bellman optimality over entire sequences, these approaches provide a powerful mechanism for LLM reasoning and alignment without requiring generative access to the environment (Wang et al., 2025). However, TBRM assumes deterministic dynamics, a condition violated in practical agentic environments where user responses and tool outputs are inherently stochastic. This discrepancy exposes a critical theoretical gap, leading to severe instability and statistical biases.

Double-sampling and debiased reinforcement learning. The statistical bias encountered when applying Bellman Residual Minimization (BRM) to stochastic environments is rooted in the "double-sampling problem" (Baird, 1995). When minimizing the squared Bellman error, the variance of the environment's stochastic transitions becomes entangled with the value function's approximation error. Unbiased estimation of this squared error requires two independent samples of the next state for the same state-action pair. Offline alignment settings cannot meet this condition, because pre-collected datasets provide only one observed continuation per state-action pair. To estimate

the objective without generative access to the environment, the classical reinforcement learning literature has explored primal-dual optimization and bi-conjugate formulations, such as Fenchel duality (Dai et al., 2018; Feng et al., 2019; Zhu & Ying, 2020). By introducing an auxiliary dual variable, these formulations decouple the transition variance from the temporal difference error, enabling debiased optimization from single-sample paths. Despite their success in classical RL, these debiasing mechanisms have not yet been adapted for multi-turn LLM alignment. Our method uses this variance-correction perspective to fit conditional continuation values and construct trajectory-level reward proxies. Debiasing is exact at the population auxiliary best response; auxiliary estimation and stop-gradient errors remain distinct in the finite-data algorithm.