LEARNING PERTURBATION ROBUST POLICIES FOR LLM AGENTS WITH STABLE OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) has become an effective post-training paradigm for long-horizon large language model (LLM) agents. However, we find that the resulting policies can be sensitive to various policy perturbations, such as hidden-state noise, pruning, and quantization. In this work, we study how to improve perturbation robustness during policy optimization. We first introduce the notion of a perturbation robust policy and analyze conditions under which perturbed policy updates preserve stable monotonic improvement. Based on this analysis, we introduce Stable Perturbation-Robust Policy Optimization (SPrPO), which applies adaptive and sensitivity-aware perturbations during RL training. We evaluate SPrPO on ALFWorld and WebShop and conduct systematic experiments across multiple perturbation types and scales, showing improved perturbation robustness while maintaining stable policy optimization.

1 INTRODUCTION

RL has become an effective paradigm for post-training LLM agents on long-horizon decision making, where the model generates textual actions in interactive environments such as web navigation and embodied household tasks (Feng et al., 2024; Qi et al., 2025; Feng et al., 2025). In practice, the deployed policy is not always the exact policy that was trained: serving costs push practitioners to prune or quantize the model, and inference may run under a different numerical precision. On static, single-turn tasks such as language modeling and question answering, where the model never acts on an environment, such perturbations are largely benign: pruning and post-training quantization compress a model substantially while retaining much of its performance (Ashkboos et al., 2024; Ma et al., 2023; Frantar et al., 2023). This tolerance does not carry over to RL-trained agents. As shown in Figure 1, pruning only 20% of the feed-forward network (FFN) channels nearly collapses performance on WebShop (Yao et al., 2022). This gap makes the robustness of RL-trained LLM agents to deployment-time perturbations an important concern for their practical use.

A natural approach to improving such robustness is to inject perturbations into the policy during training, an idea widely studied in robust RL on control problems (Pinto et al., 2017; Tessler et al., 2019; Lee & Yoon, 2025). Such methods typically operate on small policy networks trained from scratch, with dense rewards and low-dimensional continuous action spaces, so that injected perturbation has relatively mild effect. In the LLM setting, this idea has recently begun to be explored. NPFT (Wang & Yang, 2025) injects random noise into quantization-sensitive weights during fine-tuning, flattening the local loss landscape and mitigating the quantization error; related noise injection has also appeared in LLM RL, though there primarily as a source of exploration (Huang et al., 2026). For long-horizon LLM agents, robustness to policy perturbation

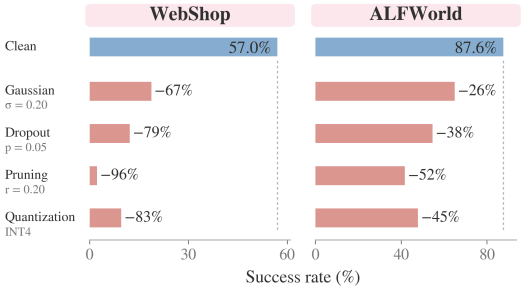


Figure 1: LLM agents trained by RL are sensitive to policy perturbations. Even with mild perturbation magnitude, success rates degrade substantially under these perturbations on both WebShop and ALFWorld. σ denotes the standard deviation for Gaussian distribution, p denotes the dropout probability and r denotes pruning ratio.

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.

remains largely unstudied, and existing methods usually target one specified type of perturbation, leaving robustness to a general family of perturbations an open and more challenging problem.

In this work, we study how to train LLM agents that remain robust to different types of policy perturbations. The difficulty is specific to the agentic setting: each action must be grounded in the environment and respect strict format constraints, and a single wrong action early in a long-horizon task can derail the whole trajectory. Therefore, perturbation injection requires careful control to preserve stability during training, and a proper perturbation scheme should avoid introducing excessive variation in policy performance. Specifically, we keep perturbations relatively mild along sensitive dimensions, since different dimensions can exhibit markedly different sensitivities to perturbations. Building on this principle, we propose **Stable Perturbation-Robust Policy Optimization (SPrPO)**, a perturbation-based method for RL training that improves the robustness of LLM agents while preserving training stability. Our main contributions are as follows:

- We are the first to study perturbation robustness for RL-trained LLM agents and formulate the problem under a general distribution of policy perturbations.
- Theoretically, we extend the monotonic policy improvement analysis of TRPO (Schulman et al., 2015) to the perturbation setting and characterize a sufficient condition for stable policy improvement. We further establish that limit points of the resulting updates are perturbation-robust.
- Guided by the theoretical analysis, we derive a practical condition for controlling perturbations during policy optimization and develop SPrPO.
- We test SPrPO on ALFWorld and WebShop under different kinds of policy perturbations, such as Gaussian noise, dropout, structured and unstructured pruning, and quantization. Across these settings, SPrPO exhibits more stable optimization during training and better preserves task performance under perturbation.

2 Related Work

We briefly review prior work on robust training, with more detailed discussion in Appendix B.

Robust Policy in Reinforcement Learning. Originating from control theory, the RL community has developed the notion of robust policies as maintaining reliable performance under disturbances (Morimoto & Doya, 2005). A broad line of work pursues this goal by deliberately introducing perturbations during training, either adversarially to improve worst-case performance (Pinto et al., 2017; Tessler et al., 2019; Mani et al., 2026) or stochastically to improve expected performance (Kumar et al., 2022; Plappert et al., 2018). Stochastic perturbations avoid explicit inner optimization and provide a simpler training mechanism. Our method takes the same idea of injecting perturbations during training, but focuses on perturbations to the policy itself rather than noise in actions or transition dynamics, which is more appropriate for LLM agents.

Perturbation-based Training for LLMs. Introducing perturbations during language-model training has been widely explored across language understanding, instruction finetuning (Jain et al., 2024), and reinforcement learning. Prior work constructs adversarial perturbations in input representations (Zhu et al., 2020; Chen et al., 2023) or through a separate low-rank parameter branch (Liu et al., 2026a). Recent RL methods also use stochastic perturbations to promote exploration and diversify reasoning trajectories (Liu et al., 2025; Jerge et al., 2026). Furthermore, perturbation-based training shows improvement on robustness to quantization, through adaptation to quantization errors (Wang & Yang, 2025) or adaptive control of quantization-induced noise during RL (Huang et al., 2026). Our work extends perturbation-based training to long-horizon LLM agents, where small policy deviations can accumulate through sequential interactions and affect future decisions. Therefore, we explicitly account for training stability, and adopt a more prudent perturbation design in practice.

3 Background and Problem Formulation

In this section, we first define the notion of perturbation robustness in Subsec. 3.1 and then review the classical TRPO lower bound and monotonic improvement theory in Subsec. 3.2.

3.1 Problem Setup and Perturbation-Robust Policy

Consider a discounted Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \rho_0, \gamma)$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ is the action space, $P(\cdot \mid s, a)$ is the transition kernel, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a bounded reward function, ρ_0 is the initial-state distribution, and $\gamma \in (0, 1)$ is the discount factor. In our setting, the agent is an LLM parameterized by θ and governed by a policy $\pi_\theta(\cdot \mid s)$. At time step t, the agent observes a state $s_t \in \mathcal{S}$, which contains the task context and interaction history available to the agent, and autoregressively generates a textual action $a_t \in \mathcal{A}$ according to $\pi_\theta(\cdot \mid s_t)$. The environment then returns a reward $r_t = r(s_t, a_t)$ and transitions to the next state $s_{t+1} \sim P(\cdot \mid s_t, a_t)$. The policy π_θ and transition kernel P induce a trajectory $\tau = (s_0, a_0, s_1, a_1, \ldots)$, and the expected discounted return of π_θ is defined as

$$J(\pi_\theta) := \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right]. \tag{1}$$

Let Ξ denote the space of perturbations and $\mathcal{P}_\xi$ the perturbation operator associated with $\xi \in \Xi$. Applying $\mathcal{P}_\xi$ to a policy π_θ induces the perturbed policy $\pi_\theta^\xi := \mathcal{P}_\xi(\pi_\theta)$. In practice, such perturbations can be applied to policy parameters or hidden representations, including Gaussian noise, random dropping, pruning, and quantization. Let $\rho(\xi)$ measure the magnitude of a perturbation ξ, and let $\Xi_\delta := \{\xi \in \Xi : \rho(\xi) \le \delta\}$ denote the set of perturbations within radius δ. We define perturbation robust policy as follows.

Definition 1 (Perturbation robustness). *Let $\nu = \nu_\delta$ be a perturbation distribution supported on Ξ_δ, and denote the expected perturbed return by $\bar{J}_{\nu_\delta}(\theta) := \mathbb{E}_{\xi \sim \nu_\delta}[J(\pi_\theta^\xi)]$. For $\epsilon \ge 0$ and $\beta \in [0, 1]$, a policy π_θ is $(\delta, \epsilon, \beta)$-probabilistic perturbation robust under ν_δ if*

$$\Pr_{\xi \sim \nu_\delta} \left(J(\pi_\theta^\xi) \ge \bar{J}_{\nu_\delta}(\theta) - \epsilon \right) \ge 1 - \beta. \tag{2}$$

Moreover, it is (δ, ϵ)-uniform perturbation robust if for any $\xi \in \Xi_\delta$, we have $J(\pi_\theta^\xi) \ge \bar{J}_{\nu_\delta}(\theta) - \epsilon$.

Uniform perturbation robustness is stronger, but can be overly restrictive in practice. Instead, probabilistic robustness provides a more practical criterion: under perturbations of magnitude at most δ, we allow a performance degradation larger than ϵ only with a small probability β, which is the objective we seek to achieve in this work.

3.2 TRPO Lower Bound and Monotonic Improvement Property

Consider one step policy update from θ to θ', whose improvement is measured by $\Delta J := J(\pi_{\theta'}) - J(\pi_\theta)$. Classical trust region policy improvement analysis (Kakade & Langford, 2002; Schulman et al., 2015) studies conditions that guarantee positive improvement, i.e., $\Delta J > 0$. The following lemma, as a central result in TRPO analysis, provides a lower bound on the policy improvement ΔJ.

Lemma 1 (TRPO lower bound (Schulman et al., 2015, Theorem 1)). *For a policy π, let $d_\pi(s) = (1 - \gamma) \sum_{t \ge 0} \gamma^t \Pr_\pi(s_t = s)$ denote its normalized discounted state visitation distribution, and let $D_{\mathrm{TV}}(p, q) = \frac{1}{2} \sum_a |p(a) - q(a)|$. For any two policies π, π' satisfying $\sup_{s,a} |A_\pi(s, a)| \le A_{\max}$, define $\alpha = \sup_s D_{\mathrm{TV}}(\pi(\cdot \mid s), \pi'(\cdot \mid s))$. Then*

$$\Delta J(\pi', \pi) \ge \frac{1}{1 - \gamma} \mathbb{E}_{s \sim d_\pi,\, a \sim \pi'(\cdot|s)}[A_\pi(s, a)] - \frac{4\gamma A_{\max}}{(1 - \gamma)^2} \alpha^2. \tag{3}$$

Lemma 1 gives the standard trust region interpretation of policy improvement. Given the current policy π_θ, denote the first term by $\mathcal{L}(\theta')$, where θ' parameterizes a candidate updated policy. This term measures the surrogate improvement relative to π_θ, while the second term penalizes excessive policy deviation. Therefore, as long as this deviation is controlled, increasing $\mathcal{L}(\theta')$ guarantees monotonic improvement in the true return.

4 Theoretical Analysis

In this section, we extend the classical monotonic improvement analysis to the perturbation setting. Subsec. 4.1 establishes stable policy improvement under perturbations and derives a practical condi-

tion for perturbation allocation. Subsec. 4.2 further shows that, under suitable regularity conditions, the resulting policy updates admit perturbation-robust limit points.

4.1 Policy Improvement under Perturbations

For perturbation ξ, let π_θ^ξ and $\pi_{\theta'}^\xi$ denote the perturbed policies before and after the update, respectively. We define the perturbed policy improvement as

$$\Delta J_\xi = \Delta J(\pi_{\theta'}^\xi; \pi_\theta^\xi) := J(\pi_{\theta'}^\xi) - J(\pi_\theta^\xi). \tag{4}$$

The quantity ΔJ_ξ therefore describes how the benefit of the update varies with perturbation, with $\Delta J_0 = \Delta J$ recovering the standard policy improvement. Now we define the expected perturbed surrogate improvement as follows.

Definition 2 (Surrogate for perturbed policy improvement)**.** *For the current policy parameter θ and a candidate updated parameter θ', we define the policy improvement surrogate under a perturbation ξ as*

$$\mathcal{L}_\xi(\theta') := \frac{1}{1-\gamma} \mathbb{E}_{\substack{s\sim d_{\pi_\theta^\xi} \\ a\sim\pi_\theta^\xi(\cdot|s)}} \left[\frac{\pi_{\theta'}^\xi(a \mid s)}{\pi_\theta^\xi(a \mid s)} A_{\pi_\theta^\xi}(s,a) \right]. \tag{5}$$

Then $\mathcal{L}_\nu(\theta') := \mathbb{E}_{\xi\sim\nu}[\mathcal{L}_\xi(\theta')]$ denotes the expected policy improvement surrogate, and the corresponding policy deviation is $\alpha_\xi(\theta') := \sup_s D_{\mathrm{TV}}\Big(\pi_\theta^\xi(\cdot \mid s), \pi_{\theta'}^\xi(\cdot \mid s)\Big)$.

We emphasize that $\mathcal{L}_\nu(\theta')$ is estimated under the state visitation distribution of the perturbed current policy π_θ^ξ, so optimizing it promotes policy improvement in expectation over perturbations. However, a positive expected improvement does not preclude degradation under a non-negligible subset of perturbations, as the realized improvement can vary substantially across perturbation realizations. Therefore, we seek a stronger stability guarantee: under the perturbation distribution, the updated policy should improve with high probability.

Theorem 1 (Stable policy improvement)**.** *Suppose that $\sup_{s,a} |A_{\pi_\theta^\xi}(s,a)| \le A_{\max}$ for ν-almost every ξ, and that $\mathrm{Var}_\nu[\Delta J_\xi] \le v_\nu < \infty$. Define the expected improvement lower bound*

$$m_\nu := \mathcal{L}_\nu(\theta') - \frac{4\gamma A_{\max}}{(1-\gamma)^2} \mathbb{E}_{\xi\sim\nu}[\alpha_\xi^2]. \tag{6}$$

Then, for any $\beta \in (0,1)$,

$$\Pr_{\xi\sim\nu}\left(\Delta J_\xi \ge m_\nu - \sqrt{\frac{1-\beta}{\beta}\, v_\nu} \right) \ge 1-\beta. \tag{7}$$

Consequently, if $m_\nu \ge \sqrt{(1-\beta)v_\nu/\beta}$, then $\Pr_{\xi\sim\nu}(\Delta J_\xi \ge 0) \ge 1-\beta$.

Theorem 1 shows that stable policy improvement depends not only on the expected improvement, but also on the variance induced by perturbations. As is illustrated in Figure 2, perturbations along sensitive directions can divert the optimization trajectory toward low-reward regions, motivating a practical way to better control variance during training. However, the perturbation distribution encountered at deployment is generally unknown, making the improvement variance difficult to characterize directly. Motivated by a common modeling principle in distributionally robust optimization, where tractable nominal distributions are used as reference for uncertain and potentially non-Gaussian disturbances (Shafieezadeh-Abadeh et al., 2018; Taskesen et al., 2023), we use Gaussian perturbations as a tractable proxy during training. Under this choice, the variance admits an explicit sensitivity-based control through dimension-dependent perturbation scales, leading to the following corollary.

Corollary 1 (Stable policy improvement under Gaussian perturbations)**.** *Under the setting of Theorem 1, let $\nu = \mathcal{N}(0,\Sigma)$ with $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_d^2)$. Suppose that $\mathcal{L}_\xi(\theta')$ is continuously differentiable with respect to ξ. For each perturbation dimension i, define the sensitivity*

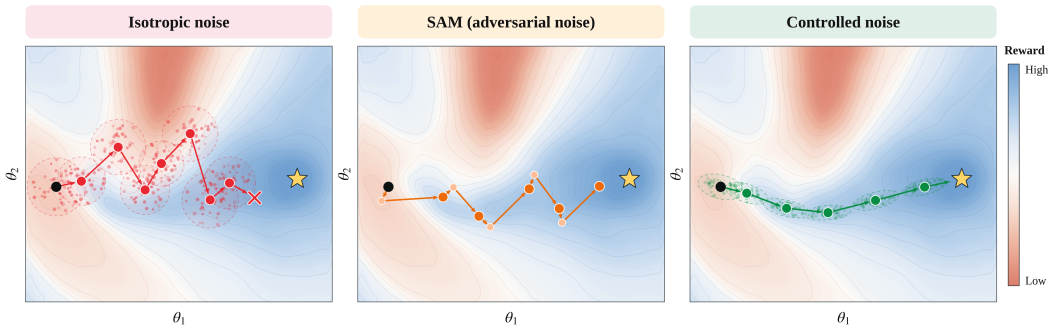


Figure 2: Illustrative example: different perturbation strategies can lead to distinct training dynamics during policy optimization. Isotropic noise (left) applies the same perturbation scale across directions, whereas SAM-style perturbations (middle) adversarially push the policy along the sensitive direction. Both can disrupt the optimization trajectory and undermine training stability. Controlled noise (right) instead limits perturbation magnitude along sensitive directions to reduce the induced variance in policy improvement and promote a more stable training process.

as $S_i(\theta') := \mathbb{E}_{\xi\sim\nu}[|\partial\mathcal{L}_\xi(\theta')/\partial\xi_i|^2] < \infty$, *and the supremum policy drift as* $\bar{\alpha} := \sup_\xi \alpha_\xi$. *If* $\mathcal{L}_\nu(\theta') - \frac{4\gamma A_{\max}}{(1-\gamma)^2}\bar{\alpha}^2 > 0$ *and the following stability condition holds:*

$$\sum_{i=1}^{d} \sigma_i^2 S_i(\theta') \le \frac{\beta}{1-\beta}\left(\mathcal{L}_\nu(\theta') - \frac{4\gamma A_{\max}}{(1-\gamma)^2}\bar{\alpha}^2\right)^2, \tag{8}$$

then $\Pr_{\xi\sim\nu}(\Delta J_\xi \ge 0) \ge 1-\beta$.

We note that the condition in Corollary 1 gives a tractable characterization of Gaussian perturbations that admit stable policy improvement. In particular, the guarantee is jointly determined by the surrogate improvement, the policy drift, and the perturbation-induced variance. This suggests that desirable perturbations should allocate smaller variance to more sensitive dimensions, and can tolerate larger variance when the surrogate improvement is larger.

4.2 Convergence under Stable Policy Updates

We next establish convergence under stable policy updates. Given $\epsilon > 0$ and $\bar{\beta} \in (0,1)$, we aim to obtain a policy whose perturbed return falls more than ϵ below its mean with probability at most $\bar{\beta}$. Let $C_J := J_{\max} - J_{\min}$ denote the range of the bounded return. Assume that the initial policy has sufficiently small return variance when excluding a negligible subset.

Assumption 1 (Proper initial variance)**.** *For the initial policy* $\pi_0 \in \Pi$, *there exists a measurable subset* $\Xi_0 \subseteq \Xi_\delta$ *with* $\nu(\Xi_0) \ge 1-\beta_0$ *such that* $\mathrm{Var}_{\xi\sim\nu(\cdot|\Xi_0)}[J(\pi_0^\xi)] \le \sigma_0^2$, *where* $\beta_0 \in [0,1)$ *and* $\sigma_0 \ge 0$ *satisfy* $\beta_0 \le \frac{1}{2}\min\{\bar{\beta}, \epsilon/C_J\}$ *and* $\sigma_0 < \frac{1}{4}\epsilon\sqrt{\bar{\beta}}$.

At iteration k, let $m_{\nu,k}(\pi')$ denote the expected improvement lower bound for a candidate policy π' relative to the current policy π_k. Following the stability condition in Theorem 1, define

$$\mathcal{U}_\beta(\pi_k) := \left\{\pi' \in \Pi :\ m_{\nu,k}(\pi') \ge \sqrt{\frac{1-\beta}{\beta}\,\mathrm{Var}_{\xi\sim\nu}\left[\Delta J\big((\pi')^\xi; \pi_k^\xi\big)\right]}\right\}. \tag{9}$$

Starting from π_0, update the policy by selecting the best $\pi_{k+1} \in \mathcal{U}_\beta(\pi_k)$ at each iteration. The following theorem shows that, as long as the probability of degradation at each step is sufficiently small, stable policy updates can lead to perturbation robust policies.

Theorem 2 (Convergence of stable policy optimization)**.** *Suppose that the policy space* Π *is compact and that* $\pi \mapsto J(\pi^\xi)$ *is continuous for* ν*-almost every* ξ. *Under Assumption 1, if* $\beta \le \epsilon^2\bar{\beta}/(100C_J^2 + \epsilon^2\bar{\beta})$, *the policy sequence* $\{\pi_k\}_{k\ge0}$ *has a nonempty set of limit points. Moreover, every limit point* $\bar{\pi}$ *is* $(\delta, \epsilon, \bar{\beta})$*-probabilistic perturbation robust.*

5 Practical Algorithm for Stable Perturbation-Robust Policy Optimization

Building on the preceding analysis, we develop a practical realization of the stability condition for LLM policy optimization. Specifically, we apply controlled Gaussian perturbations in the hidden space and estimate the sensitivity and improvement statistics from tractable rollout samples, leading to **Stable Perturbation-Robust Policy Optimization (SPrPO)**, as summarized in Algorithm 1.

Multiplicative Gaussian perturbations to FFN channels. In modern Transformer architectures, feed-forward networks (FFNs) account for a large fraction of the model parameters, operate independently on each token representation, and have been shown to store rich textual and semantic patterns (Vaswani et al., 2017; Geva et al., 2021; Yang et al., 2024). In the Qwen2.5 architecture, each FFN consists of an up-projection, a gated SiLU activation, and a down projection. To be specific, given an input x_ℓ in layer ℓ, it is computed as

$$h_\ell = \mathrm{SiLU}(W_{\mathrm{gate},\ell} x_\ell) \odot W_{\mathrm{up},\ell} x_\ell, \qquad y_\ell = W_{\mathrm{down},\ell} h_\ell,$$

Instead of modifying the full parameter space, we perturb the intermediate FFN channel h_ℓ with multiplicative Gaussian noise ξ_ℓ, which has been shown to be an effective mechanism for improving robustness (Trinh et al., 2024). Let $\xi_{\ell c} \sim \mathcal{N}(0, \sigma_{\ell c}^2)$ denote the Gaussian perturbation for channel c, i.e., coordinate c of h_ℓ, the perturbed FFN forward pass is $\widetilde{h}_{\ell c} = (1 + \xi_{\ell c}) h_{\ell c}$. Since ξ_ℓ is applied after the up projection and activation, it is equivalent to perturbing the corresponding columns of down projection matrix $W_{\mathrm{down},\ell}$, while offering a simpler implementation and more stable training.

Adaptive and sensitivity-aware Gaussian perturbations. Motivated by the stability condition in Corollary 1, we adopt non-isotropic Gaussian perturbations, i.e., $\sigma_{\ell c}$ are non-identical across different channels. However, unlike supervised fine-tuning, where loss-related statistics can be evaluated on a fixed training set (Lee et al., 2019; Wang & Yang, 2025), the exact channel sensitivities and expected improvement in reinforcement learning are difficult to evaluate because the data distribution changes with the policy. Therefore in practice, we estimate the quantities by reusing the rollout batch from the current policy update. At iteration k, we collect trajectories $\mathcal{D}_k \sim \pi_{\theta_k}^{\xi_k}$ and estimate advantages A_i for samples $i \in \mathcal{D}_k$, and optimize the policy using perturbed loss surrogate with PPO clipping

$$\mathcal{L}_k(\theta; \xi_k) = \frac{1}{|\mathcal{D}_k|} \sum_{i \in \mathcal{D}_k} \min\left(r_i(\theta) A_i,\ \mathrm{clip}\big(r_i(\theta), 1-\epsilon, 1+\epsilon\big) A_i\right),$$

where $r_i(\theta) = \pi_\theta^{\xi_k}(a_i \mid s_i) / \pi_{\theta_k}^{\xi_k}(a_i \mid s_i)$. After obtaining θ_{k+1}, we replay $\mathcal{D}_k$ for one additional forward and backward pass without extra rollouts, and estimate the channel sensitivity and expected improvement as

$$\widehat{S}_{k+1,\ell c} = \left| \frac{\partial \mathcal{L}_k(\theta_{k+1}; \xi_k)}{\partial \xi_{k,\ell c}} \right|^2, \qquad \widehat{m}_{k+1} = \|\nabla_\theta \mathcal{L}_k(\theta_{k+1}; \xi_k)\|_2^2 .$$

Algorithm 1 Stable Perturbation-Robust Policy Optimization (SPrPO)

Require: Initial policy θ_1, iterations K, base perturbation ρ, sensitivity exponent α

1: Initialize $\widehat{S}_{1,\ell c} \leftarrow 1$ and $\widehat{m}_1 \leftarrow 1$
2: **for** $k = 1, \ldots, K$ **do**
3: $\quad \sigma_{k,\ell c} \leftarrow \rho\, \widehat{m}_k\, \widehat{S}_{k,\ell c}^{-\alpha/2}$
4: $\quad$ Sample $\xi_{k,\ell c} \sim \mathcal{N}(0, \sigma_{k,\ell c}^2)$
5: $\quad$ Collect $\mathcal{D}_k \sim \pi_{\theta_k}^{\xi_k}$ and estimate advantages A_i for $i \in \mathcal{D}_k$
6: $\quad$ Update the policy: $\theta_{k+1} \leftarrow \theta_k + \eta_k \nabla_\theta \mathcal{L}_k(\theta_k; \xi_k)$
7: $\quad \widehat{S}_{k+1,\ell c} \leftarrow \left| \frac{\partial \mathcal{L}_k(\theta_{k+1}; \xi_k)}{\partial \xi_{k,\ell c}} \right|^2, \quad \widehat{m}_{k+1} \leftarrow \|\nabla_\theta \mathcal{L}_k(\theta_{k+1}; \xi_k)\|_2^2$
8: **end for**
9: **return** θ_{K+1}

Here, $\widehat{S}_{k+1,\ell c}$ measures the local sensitivity of channel (ℓ, c). For the expected improvement $\widehat{m}_{k+1}$, we use the squared PPO gradient norm as a practical proxy (Fay et al., 2023; Zhao et al., 2022), since under a small gradient ascent step, the first-order increase is proportional to the squared gradient norm. Finally, we choose standard deviation of channel perturbation to increase with the estimated improvement and decrease with channel sensitivity, i.e., $\sigma_{k+1,\ell c} = \rho\, \widehat{m}_{k+1}\, \widehat{S}_{k+1,\ell c}^{-\alpha/2}$, which follows the allocation principle implied by the stability condition equation 8. Notably, the hyperparameter $\rho > 0$ controls the base perturbation scale, while $\alpha \geq 0$ controls the strength of sensitivity adaptation: $\alpha = 0$ reduces to sensitivity-independent noise, whereas $\alpha = 1$ corresponds to inverse-square-root sensitivity scaling.

6 Experiments

In this section, we empirically validate the effectiveness of SPrPO. Subsec. 6.2 evaluates robustness under diverse policy perturbations, Subsec. 6.3 visualizes the reward landscape, and Subsec. 6.4 studies the effects of the main algorithmic components and hyperparameters. Full details of experiment settings are provided in Appendix C.

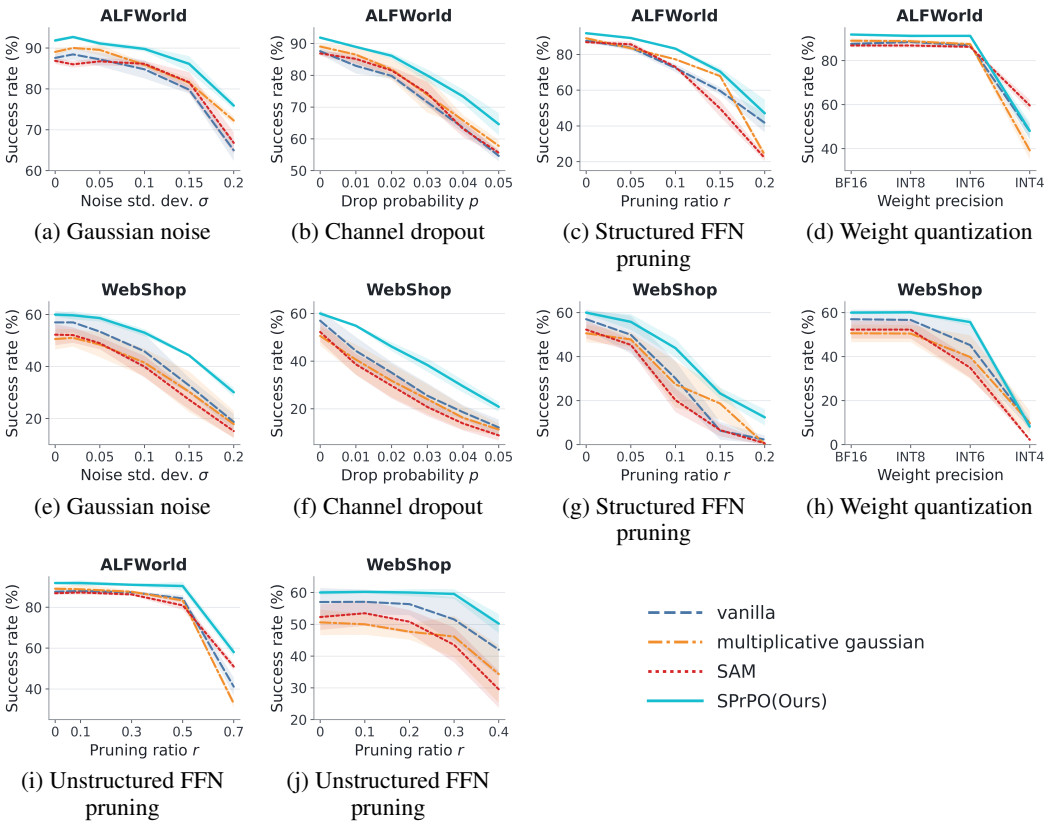
6.1 Experiment Setup

Environments. We train and evaluate the LLM agent on two commonly used long-horizon benchmarks, ALFWorld (Shridhar et al., 2021) and WebShop (Yao et al., 2022). ALFWorld is a text-based embodied environment, where an agent takes several steps to complete household tasks. WebShop is an interactive web shopping environment in which an agent searches for and purchases products according to the given instructions. In both environments, the agent receives textual observations and executes sequences of structured actions under strict environment parsers, with ALFWorld allowing up to 50 steps per episode and WebShop up to 15 steps.

Baselines. For all experiments, we use Qwen2.5-1.5B-Instruct (Yang et al., 2024) as the base model and GiGPO (Feng et al., 2025) as the base training algorithm, which is a critic-free RL method designed to better handle sparse trajectory-level rewards. We refer to the version without training-time perturbations as **Vanilla**. To our knowledge, we are the first to explicitly study policy perturbations during RL training for long-horizon LLM agents, while prior robust RL methods for continuous control do not directly transfer to this setting. We therefore consider two representative perturbation baselines: **Multiplicative Gaussian**, which applies same Gaussian variance across all channels, and **SAM** (Foret et al., 2021), which uses adversarial perturbations along the PPO loss increasing direction. Both are applied to FFN channel space in a multiplicative manner as our method, providing controlled comparisons between different perturbation strategies.

Evaluation protocol under perturbations. We evaluate agent performance by task success rate under different policy perturbations. For WebShop, we evaluate on a set of 500 shopping tasks. For ALFWorld, we evaluate on all 274 validation tasks, including 140 in-distribution (ID) tasks and 134 out-of-distribution (OOD) tasks. We consider several representative perturbations that differ in both structure and granularity:

- **Channel-wise perturbations.** These perturbations operate at the level of FFN hidden channels, which is equivalent to modifying entire rows or columns of the corresponding weight matrices. *Gaussian perturbation* applies multiplicative noise $\mathcal{N}(0, \sigma^2)$ in the same manner as during training. *Dropout* (Srivastava et al., 2014) randomly removes FFN channels with probability p. Similarly, *structured pruning* removes entire FFN channels based on their Fisher importance (Molchanov et al., 2019). Specifically, we use the same calibration set for all evaluated methods for fair comparison and estimate $\mathbb{E}[(h_{\ell,c}\, \partial \mathcal{L}/\partial h_{\ell,c})^2]$ as the importance score.
- **Weight space perturbations.** Unlike channel-wise perturbations, these perturbations operate at the level of individual weights, with a different pattern from the channel-wise perturbations used during training. We use SparseGPT (Frantar & Alistarh, 2023) for *unstructured pruning*, which uses second-order information to identify and compensate pruned weights, and apply it only to the FFN linear weights. For *quantization*, we use round-to-nearest (RTN) weight quantization (Nagel et al., 2021) over all transformer linear weights at different bit widths.



(a) Gaussian noise (b) Channel dropout (c) Structured FFN pruning (d) Weight quantization

(e) Gaussian noise (f) Channel dropout (g) Structured FFN pruning (h) Weight quantization

(i) Unstructured FFN pruning (j) Unstructured FFN pruning

Figure 3: Evaluation of perturbation robustness on ALFWorld and WebShop across increasing perturbation magnitudes. The x-axis denotes perturbation strength, measured by Gaussian standard deviation σ, dropout probability p, pruning ratio r, and quantization bit-width. SPrPO generally exhibits less severe performance degradation and maintains higher success rates.

Table 1: Evaluation of perturbation robustness on WebShop and ALFWorld. Results are reported as success rate$_{\text{std}}$ (retention) over three seeds, where retention denotes the percentage of performance preserved relative to the corresponding clean policy. SPrPO achieves the strongest performance, maintaining both higher absolute success rates and higher relative retention.

WebShop	Clean	Gaussian ($\sigma = .20$)	Dropout ($p = .05$)	Structured Pruning ($r = .20$)	Unstructured Pruning ($r = .40$)	Quantization (INT6)
Vanilla	$57.00_{\pm 4.39}$	$18.73_{\pm 3.03}$ (32.9)	$12.20_{\pm 3.30}$ (21.4)	$2.27_{\pm 1.86}$ (4.0)	$42.00_{\pm 8.31}$ (73.7)	$45.20_{\pm 9.18}$ (79.3)
Gaussian	$50.60_{\pm 3.99}$	$17.87_{\pm 5.12}$ (35.3)	$11.33_{\pm 2.60}$ (22.4)	$0.27_{\pm 0.12}$ (0.5)	$34.33_{\pm 9.00}$ (67.9)	$39.80_{\pm 9.90}$ (78.7)
SAM	$52.27_{\pm 4.02}$	$15.20_{\pm 2.65}$ (29.1)	$8.87_{\pm 2.14}$ (17.0)	$0.60_{\pm 0.87}$ (1.1)	$29.53_{\pm 5.74}$ (56.5)	$34.93_{\pm 4.54}$ (66.8)
SPrPO (Ours)	$\mathbf{60.00}_{\pm 0.72}$	$\mathbf{30.13}_{\pm 1.81}$ (50.2)	$\mathbf{20.87}_{\pm 2.00}$ (34.8)	$\mathbf{12.40}_{\pm 3.65}$ (20.7)	$\mathbf{50.20}_{\pm 2.99}$ (83.7)	$\mathbf{55.67}_{\pm 1.60}$ (92.8)
ALFWorld	Clean	Gaussian ($\sigma = .20$)	Dropout ($p = .05$)	Structured Pruning ($r = .20$)	Unstructured Pruning ($r = .70$)	Quantization (INT6)
Vanilla	$87.59_{\pm 0.97}$	$64.96_{\pm 2.55}$ (74.2)	$54.62_{\pm 1.65}$ (62.4)	$41.73_{\pm 5.32}$ (47.6)	$41.12_{\pm 4.35}$ (46.9)	$86.98_{\pm 0.84}$ (99.3)
Gaussian	$89.05_{\pm 0.97}$	$72.26_{\pm 0.97}$ (81.1)	$57.79_{\pm 1.80}$ (64.9)	$23.72_{\pm 2.28}$ (26.6)	$32.85_{\pm 1.93}$ (36.9)	$87.47_{\pm 0.92}$ (98.2)
SAM	$86.86_{\pm 0.73}$	$66.79_{\pm 2.55}$ (76.9)	$55.60_{\pm 0.76}$ (64.0)	$22.02_{\pm 1.69}$ (25.4)	$50.97_{\pm 1.69}$ (58.7)	$86.25_{\pm 0.56}$ (99.3)
SPrPO (Ours)	$\mathbf{91.85}_{\pm 0.42}$	$\mathbf{75.91}_{\pm 0.97}$ (82.6)	$\mathbf{64.60}_{\pm 3.48}$ (70.3)	$\mathbf{47.08}_{\pm 7.65}$ (51.3)	$\mathbf{58.03}_{\pm 2.39}$ (63.2)	$\mathbf{91.24}_{\pm 0.00}$ (99.3)

6.2 Main Results: Robustness to Diverse Policy Perturbations

We evaluate the robustness of the policies under diverse perturbations. Figure 3 reports the full robustness curves as the perturbation strength increases, while Table 1 summarizes the results under relatively strong perturbation settings for direct comparison. Overall, policies trained by our method are generally more robust than all baselines. At sufficiently extreme perturbation levels, however, all methods can eventually collapse. For example, under INT4 quantization on WebShop, success rates

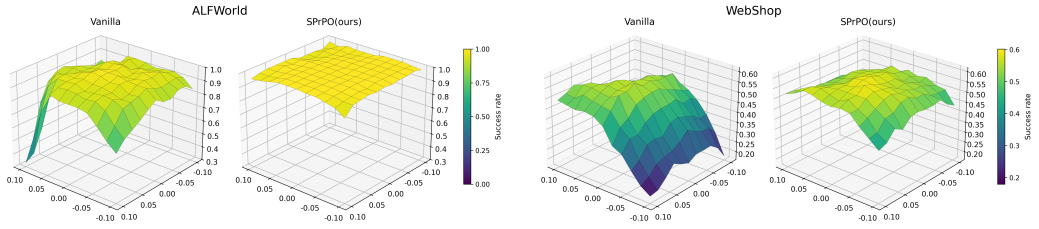


Figure 4: Reward landscape of policies trained with Vanilla and SPrPO on ALFWorld and WebShop. Vanilla exhibits sharp performance cliffs, while SPrPO shows flatter reward surfaces.

fall to the single digits, where differences between policies become less informative and performance is close to the failure regime.

6.3 Reward Landscape Visualization

To further examine local sensitivity to perturbations, we visualize reward surfaces, following prior work on reward landscape analysis (Sullivan et al., 2022; Lee & Yoon, 2025). For each trained policy, we sample two fixed multiplicative Gaussian perturbation directions, ξ_1 and ξ_2, and vary their magnitudes w_1 and w_2 over a two-dimensional grid. Each point corresponds to the policy under perturbation $w_1\xi_1 + w_2\xi_2$, with the z-axis representing the success rate, as shown in Figure 4.

6.4 Ablation and Hyperparameter Study

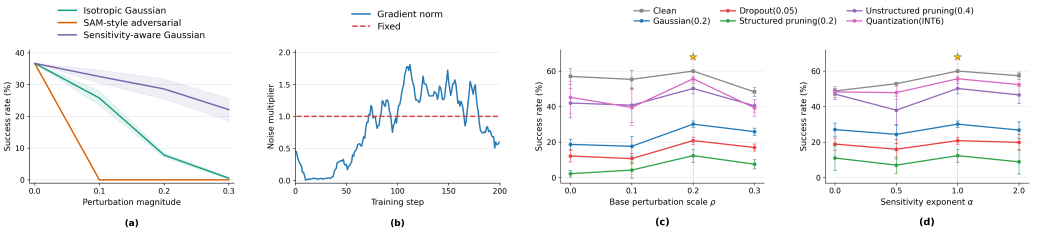


Figure 5: Ablation and hyperparameter study. (a) Comparison of perturbation strategies during training. (b) Effect of gradient-norm scaling. (c,d) Effects of perturbation scale ρ and sensitivity exponent α. Starred and $\rho = 0$ results reuse the main-experiment estimates in Table 1.

Figure 5(a,b) examines the two main design choices of SPrPO. Under the same perturbation magnitude, SAM-style perturbations collapse task performance, and isotropic Gaussian noise also causes substantial degradation. In contrast, reducing perturbations on sensitive channels preserves more original performance. Since SAM is substantially more aggressive at the same magnitude, we further reduce its perturbation scale to obtain a stable training baseline. Gradient-norm scaling increases noise strength as training progresses; removing it reduces the clean success rate by 7.87 percentage points (Table 9).

Figure 5(c,d) shows the effect of hyperparameters ρ and α. A small ρ provides insufficient perturbation exposure to improve robustness, whereas an overly large ρ corrupts the training signal and causes the model to underfit. For α, a small value does not provide enough protection, leaving sensitive channels over-perturbed, and further increasing α also reduces the robustness gains, since sensitive channels become barely perturbed during training.

7 Conclusion and Future Work

We study perturbation robustness in RL-trained LLM agents and theoretically characterize policy improvement under perturbations. Building on this analysis, we develop SPrPO and show that it improves robustness across diverse policy perturbations. For future work, further analysis could consider different model structures, agent tasks, and perturbation settings.

AI USE STATEMENT

We used generative AI tools to assist with algorithm implementation and language polishing during the preparation of this work. All algorithmic design, theoretical analysis, experimental design, results, and conclusions were developed and verified by the authors. The authors take full responsibility for the content of the paper.

REPRODUCIBILITY STATEMENT

Complete proofs of the theoretical results are included in Appendix A. Appendix C provides detailed descriptions of the environments, training procedure, policy perturbations, algorithm implementation, and hyperparameters used in our experiments.

REFERENCES

Richard Aoun, Marwa Banna, and Pierre Youssef. Matrix poincaré inequalities and concentration. *Advances in Mathematics*, 371:107251, 2020. ISSN 0001-8708. doi: https://doi.org/10.1016/j.aim.2020.107251. URL https://www.sciencedirect.com/science/article/pii/S0001870820302772.

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. SliceGPT: Compress large language models by deleting rows and columns. In *International Conference on Learning Representations*, 2024.

Lichang Chen, Jiuhai Chen, Heng Huang, and Minhao Cheng. PTP: Boosting stability and performance of prompt tuning with perturbation-based regularizer. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=9rWqOgvGpc.

Dominik Fay, Sindri Magnússon, Jens Sjölund, and Mikael Johansson. Adaptive hyperparameter selection for differentially private gradient descent. *Transactions on Machine Learning Research*, 2023.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=QXEhBMNrCW.

Peiyuan Feng, Yichen He, Guanhua Huang, Yuan Lin, Hanchong Zhang, Yuchen Zhang, and Hang Li. AGILE: A novel reinforcement learning framework of LLM agents. In *Advances in Neural Information Processing Systems*, 2024.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM.

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tcbBPnfwxS.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, 2021.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Wei Huang, Yi Ge, Shuai Yang, Yicheng Xiao, Huizi Mao, Yujun Lin, Hanrong Ye, Sifei Liu, Ka Chun Cheung, Hongxu Yin, Yao Lu, Xiaojuan Qi, Song Han, and Yukang Chen. QeRL: Beyond efficiency—quantization-enhanced reinforcement learning for llms. In *International Conference on Learning Representations*, 2026.

Neel Jain, Ping-yeh Chiang, Yuxin Wen, John Kirchenbauer, Hong-Min Chu, Gowthami Somepalli, Brian R. Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. NEFTune: Noisy embeddings improve instruction finetuning. In *International Conference on Learning Representations*, 2024.

Michael Jerge, Joseph Pelczar, and Justin Downes. Perturb the thought, not the pixels: Latent-space rollout diversification for reinforcement learning of vision-language models. *arXiv preprint arXiv:2608.21595*, 2026.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pp. 267–274, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. ISBN 1558608737.

Aounon Kumar, Alexander Levine, and Soheil Feizi. Policy smoothing for provably robust reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=mwdfai8NBrJ.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, 2023.

Hyun Kyu Lee and Sung Whan Yoon. Flat reward in policy parameter space implies robust reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4OaO3GjP7k.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Snip: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2019.

Xiangyan Liu, Jinjie Ni, Zijian Wu, Chao Du, Longxu Dou, Haonan Wang, Tianyu Pang, and Michael Qizhe Shieh. Noisyrollout: Reinforcing visual reasoning with data augmentation. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Yuhang Liu, Tao Li, Zhehao Huang, Zuopeng Yang, and Xiaolin Huang. Bi-lora: Efficient sharpness-aware minimization for fine-tuning large-scale models. In *International Conference on Learning Representations*, 2026a.

Zhihan Liu, Lin Guan, Yixin Nie, Kai Zhang, Zhuoqun Hao, Lin Chen, Asli Celikyilmaz, Zhaoran Wang, and Na Zhang. Paying less generalization tax: A cross-domain generalization study of RL training for LLM agents. arXiv preprint arXiv:2601.18217, 2026b. URL https://arxiv.org/abs/2601.18217.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-Pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Kaustubh Mani, Yann Pequignot, Vincent Mai, and Liam Paull. SHAPO: Sharpness-aware policy optimization for safe exploration. In *International Conference on Learning Representations*, 2026.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11256–11264, 2019. doi: 10.1109/CVPR.2019.01152.

Jun Morimoto and Kenji Doya. Robust reinforcement learning. *Neural Comput.*, 17(2):335–359, February 2005. ISSN 0899-7667. doi: 10.1162/0899766053011528. URL https://doi.org/10.1162/0899766053011528.

Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021. URL https://arxiv.org/abs/2106.08295.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 2817–2826. JMLR.org, 2017.

Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ByBAl2eAZ.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiadai Sun, Xinyue Yang, Yu Yang, Shuntian Yao, Wei Xu, Jie Tang, and Yuxiao Dong. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 79791–79821, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/c66e1fcc9691aae706250638f36f681b-Paper-Conference.pdf.

John Schulman, Sergey Levine, Philipp Moritz, Michael Jordan, and Pieter Abbeel. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning - Volume 37*, ICML'15, pp. 1889–1897. JMLR.org, 2015.

Soroosh Shafieezadeh-Abadeh, Viet Anh Nguyen, Daniel Kuhn, and Peyman Mohajerin Esfahani. Wasserstein distributionally robust kalman filtering. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 8483–8492, Red Hook, NY, USA, 2018. Curran Associates Inc.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1): 1929–1958, January 2014. ISSN 1532-4435.

Ryan Sullivan, Jordan K. Terry, Benjamin Black, and John P. Dickerson. Cliff diving: Exploring reward surfaces in reinforcement learning environments. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 20744–20776. PMLR, 2022.

Bahar Taskesen, Dan A. Iancu, Çağıl Koçyiğit, and Daniel Kuhn. Distributionally robust linear quadratic control. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6215–6224. PMLR, 2019.

Trung Trinh, Markus Heinonen, Luigi Acerbi, and Samuel Kaski. Improving robustness to corruptions with multiplicative weight perturbations. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Dongwei Wang and Huanrui Yang. Taming sensitive weights: Noise perturbation fine-tuning for robust llm quantization. In *Conference on Parsimony and Learning*, volume 280 of *Proceedings of Machine Learning Research*, pp. 884–896. PMLR, 2025.

Yi-Shan Wu, Andrés R. Masegosa, Stephan S. Lorenzen, Christian Igel, and Yevgeny Seldin. Chebyshev-cantelli pac-bayes-bennett inequality for the weighted majority vote. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: towards scalable real-world web interaction with grounded language agents. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Zijing Zhang and Boning Zhang. When instinct guides and insight grounds: Staged rl training for llm agents. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence (AAAI-26)*, pp. 34906–34914, 2026.

Yang Zhao, Hao Zhang, and Xiuyuan Hu. Penalizing gradient norm for efficiently improving generalization in deep learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 26982–26992. PMLR, 2022.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. FreeLB: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BygzbyHFvB.

A Detailed Proofs for Theoretical Results

A.1 Proof of Theorem 1

Proof. For each fixed perturbation ξ, applying Lemma 1 to π_θ^ξ and $\pi_{\theta'}^\xi$ gives

$$\Delta J_\xi \geq \frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d_{\pi_\theta^\xi} \\ a\sim\pi_{\theta'}^\xi(\cdot|s)}}\left[A_{\pi_\theta^\xi}(s,a)\right] - \frac{4\gamma A_{\max}}{(1-\gamma)^2}\alpha_\xi^2.$$

Taking expectation over $\xi \sim \nu$ yields

$$\mathbb{E}_{\xi\sim\nu}[\Delta J_\xi] \geq \mathcal{L}_\nu(\theta') - \frac{4\gamma A_{\max}}{(1-\gamma)^2}\mathbb{E}_{\xi\sim\nu}[\alpha_\xi^2] = m_\nu.$$

Let $\mu_\nu := \mathbb{E}_{\xi\sim\nu}[\Delta J_\xi]$. Then $\mu_\nu \geq m_\nu$, and by assumption $\mathrm{Var}_\nu[\Delta J_\xi] \leq v_\nu$. By Cantelli's inequality (Wu et al., 2021), for any $t > 0$,

$$\Pr_{\xi\sim\nu}\left(\Delta J_\xi - \mu_\nu \leq -t\right) \leq \frac{\mathrm{Var}_\nu[\Delta J_\xi]}{\mathrm{Var}_\nu[\Delta J_\xi] + t^2}.$$

Since $x/(x+t^2)$ is increasing in $x \geq 0$, choosing $t = \sqrt{(1-\beta)v_\nu/\beta}$ gives

$$\Pr_{\xi\sim\nu}\left(\Delta J_\xi < \mu_\nu - \sqrt{\frac{1-\beta}{\beta}v_\nu}\right) \leq \frac{v_\nu}{v_\nu + \frac{1-\beta}{\beta}v_\nu} = \beta.$$

Therefore,

$$\Pr_{\xi\sim\nu}\left(\Delta J_\xi \geq m_\nu - \sqrt{\frac{1-\beta}{\beta}v_\nu}\right) \geq \Pr_{\xi\sim\nu}\left(\Delta J_\xi \geq \mu_\nu - \sqrt{\frac{1-\beta}{\beta}v_\nu}\right) \geq 1-\beta.$$

□

A.2 Proof of Corollary 1

Proof. Let $C := 4\gamma A_{\max}/(1-\gamma)^2$. For each perturbation ξ, the TRPO improvement bound gives

$$\Delta J_\xi \geq \mathcal{L}_\xi(\theta') - C\alpha_\xi^2 \geq \mathcal{L}_\xi(\theta') - C\bar{\alpha}^2.$$

Since $\xi \sim \mathcal{N}(0,\Sigma)$ and $\mathcal{L}_\xi(\theta')$ is continuously differentiable in ξ, the Gaussian Poincaré inequality (Aoun et al., 2020) gives the variance bound as follows:

$$\begin{aligned}\mathrm{Var}_{\xi\sim\nu}[\mathcal{L}_\xi(\theta')] &\leq \mathbb{E}_{\xi\sim\nu}\left[\nabla_\xi\mathcal{L}_\xi(\theta')^\top\Sigma\nabla_\xi\mathcal{L}_\xi(\theta')\right] \\ &= \sum_{i=1}^d \sigma_i^2 S_i(\theta') \\ &\leq \frac{\beta}{1-\beta}\left(\mathcal{L}_\nu(\theta') - C\bar{\alpha}^2\right)^2.\end{aligned}$$

Applying Cantelli's inequality to $\mathcal{L}_\xi(\theta')$ yields

$$\Pr_{\xi\sim\nu}\left(\mathcal{L}_\xi(\theta') - C\bar{\alpha}^2 \geq 0\right) \geq 1-\beta.$$

Since $\Delta J_\xi \geq \mathcal{L}_\xi(\theta') - C\bar{\alpha}^2$ for every ξ, this yields $\Pr_{\xi\sim\nu}(\Delta J_\xi \geq 0) \geq 1-\beta$. □

A.3 Proof of Theorem 2

Proof. **Step 1: Convergence.** Since Π is compact, every sequence in Π admits a convergent subsequence. Hence, $\{\pi_k\}_{k\geq 0}$ has at least one limit point. By construction, $\pi_{k+1} \in \mathcal{U}_\beta(\pi_k)$, which means

$$\bar{J}_{k+1} - \bar{J}_k \geq m_{\nu,k}(\pi_{k+1}) \geq \sqrt{\frac{1-\beta}{\beta}\operatorname{Var}_\nu\left[\Delta J(\pi^\xi_{k+1};\pi^\xi_k)\right]} \geq 0,$$

where $\bar{J}_k := \mathbb{E}_{\xi\sim\nu}[J(\pi^\xi_k)]$. Thus, $\{\bar{J}_k\}_{k\geq 0}$ is non-decreasing. Since J is bounded, $\{\bar{J}_k\}_{k\geq 0}$ is also bounded above, and therefore converges to a finite limit, denoted by $J^\star$. Let $\bar{\pi}$ be any limit point of $\{\pi_k\}_{k\geq 0}$, and let $\pi_{k_j} \to \bar{\pi}$ be a convergent subsequence. By continuity of the perturbed return and the dominated convergence theorem, we have

$$\mathbb{E}_{\xi\sim\nu}[J(\bar{\pi}^\xi)] = \lim_{j\to\infty}\mathbb{E}_{\xi\sim\nu}[J(\pi^\xi_{k_j})] = \lim_{j\to\infty}\bar{J}_{k_j} = J^\star.$$

Step 2: Probabilistic robustness. Let $\Xi_0 \subseteq \Xi_\delta$ be the subset in Assumption 1, and let $\nu_0 := \nu(\cdot \mid \Xi_0)$. Then $\nu(\Xi_0) \geq 1-\beta_0$ and $\sqrt{\operatorname{Var}_{\nu_0}[J(\pi^\xi_0)]} \leq \sigma_0$. For any square-integrable random variable X, conditioning on an event of probability at least $1-\beta_0$ gives $\operatorname{Var}_{\nu_0}[X] \leq (1-\beta_0)^{-1}\operatorname{Var}_\nu[X]$. Together with the defining condition of $\mathcal{U}_\beta(\pi_k)$, this yields

$$\sqrt{\operatorname{Var}_{\nu_0}\left[\Delta J(\pi^\xi_{k+1};\pi^\xi_k)\right]} \leq \frac{1}{\sqrt{1-\beta_0}}\sqrt{\frac{\beta}{1-\beta}}(\bar{J}_{k+1}-\bar{J}_k).$$

By the triangle inequality for standard deviations,

$$\begin{aligned}\sqrt{\operatorname{Var}_{\nu_0}[J(\pi^\xi_{k+1})]} - \sqrt{\operatorname{Var}_{\nu_0}[J(\pi^\xi_k)]} &\leq \sqrt{\operatorname{Var}_{\nu_0}\left[\Delta J(\pi^\xi_{k+1};\pi^\xi_k)\right]} \\ &\leq \frac{1}{\sqrt{1-\beta_0}}\sqrt{\frac{\beta}{1-\beta}}(\bar{J}_{k+1}-\bar{J}_k).\end{aligned}$$

Summing from 0 to $k-1$ gives

$$\begin{aligned}\sqrt{\operatorname{Var}_{\nu_0}[J(\pi^\xi_k)]} &\leq \sigma_0 + \frac{1}{\sqrt{1-\beta_0}}\sqrt{\frac{\beta}{1-\beta}}(\bar{J}_k - \bar{J}_0) \\ &\leq \sigma_0 + \frac{C_J}{\sqrt{1-\beta_0}}\sqrt{\frac{\beta}{1-\beta}} =: \bar{\sigma},\end{aligned}$$

where $C_J := J_{\max} - J_{\min}$ denotes the range of the bounded return. By continuity and uniform boundedness of the perturbed return, its first and second moments under ν_0 converge along the subsequence. Hence, $\sqrt{\operatorname{Var}_{\nu_0}[J(\bar{\pi}^\xi)]} \leq \bar{\sigma}$. Since $\nu_0 = \nu(\cdot \mid \Xi_0)$ with $\nu(\Xi_0) \geq 1-\beta_0$,

$$\left|\mathbb{E}_\nu[J(\bar{\pi}^\xi)] - \mathbb{E}_{\nu_0}[J(\bar{\pi}^\xi)]\right| \leq \beta_0 C_J.$$

By Assumption 1, $\beta_0 \leq \bar{\beta}/2$ and $\epsilon - \beta_0 C_J \geq \epsilon/2 > 0$. Using the preceding bound on the conditional mean and applying Cantelli's inequality, we obtain

$$\begin{aligned}\Pr_{\xi\sim\nu}\left(J(\bar{\pi}^\xi) < J^\star - \epsilon\right) &\leq \beta_0 + (1-\beta_0)\Pr_{\xi\sim\nu_0}\left(J(\bar{\pi}^\xi) < \mathbb{E}_{\xi\sim\nu_0}[J(\bar{\pi}^\xi)] - (\epsilon - \beta_0 C_J)\right) \\ &\leq \beta_0 + (1-\beta_0)\frac{\bar{\sigma}^2}{\bar{\sigma}^2 + (\epsilon-\beta_0 C_J)^2}.\end{aligned}$$

The condition $\beta \leq \epsilon^2\bar{\beta}/(100C_J^2 + \epsilon^2\bar{\beta})$ implies

$$C_J\sqrt{\frac{\beta}{1-\beta}} \leq \frac{1}{10}\epsilon\sqrt{\bar{\beta}}.$$

Together with $\sigma_0 < \epsilon\sqrt{\bar{\beta}}/4$ from Assumption 1, this gives

$$\begin{aligned}\beta_0 + (1-\beta_0)\frac{\bar{\sigma}^2}{\bar{\sigma}^2 + (\epsilon - \beta_0 C_J)^2} &\le \beta_0 + \frac{4(1-\beta_0)\bar{\sigma}^2}{\epsilon^2} \\ &= \beta_0 + \frac{4}{\epsilon^2}\left(\sqrt{1-\beta_0}\,\sigma_0 + C_J\sqrt{\frac{\beta}{1-\beta}}\right)^2 \\ &\le \frac{\bar{\beta}}{2} + 4\left(\frac{1}{4} + \frac{1}{10}\right)^2 \bar{\beta} \\ &= \frac{99}{100}\bar{\beta} \le \bar{\beta}.\end{aligned}$$

It follows that

$$\Pr_{\xi\sim\nu}\left(J(\bar{\pi}^{\xi}) \ge J^{\star} - \epsilon\right) \ge 1 - \bar{\beta},$$

which shows that every limit point $\bar{\pi}$ is $(\delta, \epsilon, \bar{\beta})$-probabilistic perturbation robust under ν_δ. □

B Additional Related Work

Robust Policy in Reinforcement Learning. There are several perturbation-based approaches to robust RL, and they have different designs for how perturbations are constructed and where they enter the interaction process. One major line constructs adversarial perturbations, including RARL (Pinto et al., 2017), Action Robust RL (Tessler et al., 2019), and SHAPO (Mani et al., 2026). These methods seek perturbations that minimize policy performance and train the policy against them, typically following a min–max objective $\max_\theta \min_{\xi\in\Xi} J(\pi_\theta^\xi)$, where the adversarial perturbation may be approximated using gradient information, closely related to SAM-style (Foret et al., 2021) optimization.

Another line uses stochastic rather than worst-case perturbations, typically optimizing an average-case objective $\max_\theta \mathbb{E}_\xi[J(\pi_\theta^\xi)]$. For example, Kumar et al. (2022) inject Gaussian noise into observations, whereas Plappert et al. (2018) perturb policy parameters. Stochastic perturbations provide a simpler training mechanism, avoiding explicit inner optimization in constructing adversarial perturbations. Our work adopts stochastic perturbations and studies their role in improving the robustness of long-horizon LLM agents.

Perturbation-based Training for LLMs. Introducing perturbations during LLM training has been widely explored across different settings for better model robustness, including natural language understanding, instruction finetuning (Jain et al., 2024), and reinforcement learning. Several studies consider adversarial perturbations. FreeLB (Zhu et al., 2020) and PTP (Chen et al., 2023) apply adversarial perturbations in the input representation space in single-turn language tasks. Also, Liu et al. (2026a) parameterize adversarial perturbations with a separate low-rank LoRA branch, which can be both more efficient and stable for LLMs.

For multi-turn or reasoning-based tasks, stochastic perturbations are more commonly used during finetuning for their simplicity and training stability. For example, Liu et al. (2025) perturb model inputs during rollouts, and Jerge et al. (2026) perturb latent representations with Gaussian noise; both promote LLM exploration and diversify reasoning trajectories, thereby improving reasoning performance. More importantly, recent work shows that perturbation-based training can help models better preserve their performance after quantization. Wang & Yang (2025) perturb sensitive components during supervised finetuning to better adapt quantization errors, while Huang et al. (2026) exploits and adaptively controls quantization-induced noise during RL to balance robustness and training stability. In contrast, we study perturbation-based training in long-horizon agentic interactions, where perturbations can alter subsequent observations over the trajectory. This motivates explicit consideration of optimization stability and more careful control of the injected perturbations.

C Experiment Details

The code will be made publicly available upon acceptance.

C.1 Environment Details

Basic setup. We conduct experiments on ALFWorld (Shridhar et al., 2021) and WebShop (Yao et al., 2022). ALFWorld is a text-based embodied environment for household tasks, with 3,553 tasks in its training split. We evaluate on all 274 validation tasks, consisting of 140 in-distribution and 134 out-of-distribution tasks. For WebShop, the benchmark contains 12,087 shopping instructions, with 500 tasks reserved for testing. We use the remaining 11,587 tasks as the RL training pool and evaluate on all 500 test tasks. Episodes terminate upon task completion or when the maximum number of environment steps is reached, which is 50 for ALFWorld and 15 for WebShop.

Observation. At each environment step, the policy receives a textual observation containing the task instruction, recent interaction history, the current environment observation, and the currently admissible actions. We use the following common prompt structure for both environments:

Prompt template

Task: {task instruction}
Interaction context: {recent observations and actions}
Current observation: {current observation}
Available actions: {admissible actions}
Response: Output the next action in `<action>...</action>`.

Action. The policy directly generates an environment action at each step. For ALFWorld, the generated content is a structured natural-language command such as `go to fridge 1`; similarly for WebShop, it follows the environment action format, such as `search[wireless headphones]` or `click[Buy Now]`.

We use a no-thinking setting (Liu et al., 2026b; Zhang & Zhang, 2026), disabling explicit reasoning traces throughout both training and evaluation. This choice is motivated by both optimization and computational considerations. First, hallucinations in intermediate reasoning can propagate to the final decision. Second, RL updates over long reasoning traces can introduce token-level credit-assignment ambiguity, since incorrect intermediate thoughts may still be reinforced by a positive trajectory-level reward. Finally, thinking-free mode is also natural for many of the grounded tasks, where actions such as navigating to or picking up an object often do not require lengthy explicit deliberation. Besides, removing explicit reasoning substantially reduces training cost in our setting, shortening training from roughly seven days to two and yielding about a **3.5x** speedup.

Reward. Both environments provide sparse trajectory-level task rewards following GiGPO (Feng et al., 2025). ALFWorld uses a sparse success-based reward. WebShop instead provides a partial-completion score based on how well the purchased product matches the target attributes. We also apply a mild step-level format penalty of -0.1 when the response does not follow the required action format, which helps accelerate policy convergence.

C.2 Training Details

Base model and optimization. We fine-tune Qwen2.5-1.5B-Instruct with LoRA (Hu et al., 2022) applied to all attention and FFN linear projections, keeping the base-model weights frozen. All methods use GiGPO (Feng et al., 2025) for advantage estimation and optimize the same PPO clipped surrogate objective. All methods share the training hyperparameters in Table 2.

Perturbation implementation. Gaussian, SAM, and SPrPO perturb the intermediate representation in each FFN block in the same manner. Let $h_\ell = \mathrm{SiLU}(W_{\mathrm{gate},\ell} x_\ell) \odot W_{\mathrm{up},\ell} x_\ell$ denote the activated FFN representation at layer ℓ before the down-projection. We apply a channel-wise multiplicative perturbation $h_\ell \leftarrow h_\ell \odot (1 + \xi_\ell)$ before computing $W_{\mathrm{down},\ell} h_\ell$. The main perturbation hyperparameters are summarized in Table 3. We also experimented with parameter-space noise and dropout during training, but both led to training collapse, consistent with our observation that RL-trained LLM agents are fragile to policy perturbations.

Table 2: Shared training hyperparameters used across all methods.

Hyperparameter	Value
Base model	Qwen2.5-1.5B-Instruct
LoRA rank / α_{LoRA}	64 / 64
Optimizer	Adam
Learning rate	3×10^{-6}
Training steps	200
Tasks per step	16
Rollouts per task	8
Trajectories per PPO batch	128
Discount factor γ	0.95
PPO clip ratio	0.2
Dual-clip coefficient	3.0
Entropy coefficient	0.001
WebShop prompt / response length	4096 / 512
ALFWorld prompt / response length	3072 / 1024
Validation temperature	0.4

Table 3: Perturbation-specific training hyperparameters used in the main experiments.

Method	Hyperparameter	Value
Gaussian	Noise scale σ	0.10
SAM	Perturbation scale ρ	0.005
SPrPO	Base perturbation scale ρ	0.20
SPrPO	Sensitivity exponent α	1.0

Baseline 1: Vanilla. Vanilla uses standard GiGPO training without policy perturbations. The resulting procedure is summarized in Algorithm 2.

Algorithm 2 Vanilla GiGPO

Require: Initial policy θ_1, iterations K
1: **for** $k = 1, \ldots, K$ **do**
2: Collect $\mathcal{D}_k \sim \pi_{\theta_k}$
3: Estimate group-relative advantages A_i for $i \in \mathcal{D}_k$
4: Update the policy: $\theta_{k+1} \leftarrow \theta_k + \eta_k \nabla_\theta \mathcal{L}_k(\theta_k)$
5: **end for**
6: **return** θ_{K+1}

Baseline 2: Multiplicative Gaussian. The Gaussian baseline uses isotropic multiplicative noise; that is, the perturbation variance is the same across all perturbed FFN channels. At each training iteration, we sample a Gaussian perturbation and keep it fixed throughout the corresponding rollout and policy update.

Baseline 3: SAM. We adapt Sharpness-Aware Minimization (Foret et al., 2021) to the same FFN-channel perturbation space used by Gaussian and SPrPO. At each training iteration, we perform one additional backward pass on the PPO loss to estimate the gradient with respect to multiplicative channel gain $g_{\ell c} = \partial \mathcal{L} / \partial u_{\ell c}|_{u_{\ell c}=1}$. We normalize this gradient to construct an adversarial perturbation, then replay the same training batch under this perturbation to update the policy. SAM is extremely sensitive in our experiment: using $\rho = 0.1$ leads to performance collapse. We therefore use a much smaller scale of $\rho = 5 \times 10^{-3}$ to obtain reasonable performance.

Our method: SPrPO. We follow Algorithm 1 in the main text. SPrPO estimates a channel-wise sensitivity statistic from the gradient of the training surrogate with respect to the multiplicative FFN perturbation. The resulting noise scale increases with the overall perturbation magnitude ρ and decreases with channel sensitivity according to exponent α, such that more sensitive channels

Algorithm 3 Multiplicative Gaussian perturbation training

Require: Initial policy θ_1, iterations K, noise scale σ
1: **for** $k = 1, \ldots, K$ **do**
2: Sample z_k, where $z_{k,\ell c} \sim \mathcal{N}(0, 1)$
3: Set $\xi_k \leftarrow \sigma z_k$
4: Collect $\mathcal{D}_k \sim \pi_{\theta_k}^{\xi_k}$ and estimate advantages A_i
5: Update the policy: $\theta_{k+1} \leftarrow \theta_k + \eta_k \nabla_\theta \mathcal{L}_k(\theta_k; \xi_k)$
6: **end for**
7: **return** θ_{K+1}

Algorithm 4 FFN-channel SAM perturbation training

Require: Initial policy θ_1, iterations K, perturbation scale ρ
1: **for** $k = 1, \ldots, K$ **do**
2: Collect $\mathcal{D}_k \sim \pi_{\theta_k}$ and estimate advantages A_i
3: Estimate channel gradients: $g_{k,\ell c} \leftarrow \left.\dfrac{\partial \mathcal{L}_k(\theta_k; \mathbf{u} - \mathbf{1})}{\partial u_{\ell c}}\right|_{\mathbf{u}=\mathbf{1}}$
4: $\xi_k \leftarrow -\rho\, g_k / \|g_k\|_2$
5: Replay $\mathcal{D}_k$ under ξ_k and update the policy: $\theta_{k+1} \leftarrow \theta_k + \eta_k \nabla_\theta \mathcal{L}_k(\theta_k; \xi_k)$
6: **end for**
7: **return** θ_{K+1}

receive smaller perturbations. After each policy update, we replay the current training batch at θ_{k+1} for one additional forward/backward pass to estimate the sensitivity statistics.

Training overhead. Table 4 reports the observed training cost of SPrPO relative to Vanilla. Experiments were run with two NVIDIA RTX PRO 6000 Blackwell Max-Q GPUs on a host equipped with AMD EPYC 9354 CPU. SPrPO introduces no additional environment rollout, and its main computational overhead comes from one additional forward/backward pass that replays the current training batch after the policy update to estimate channel sensitivity.

Table 4: Observed training cost of Vanilla and SPrPO on Qwen2.5-1.5B-Instruct. Times are averaged seconds per training step and exclude periodic validation and checkpoint saving.

Environment	Method	Rollout	Update	Total
ALFWorld	Vanilla	143.8	198.2	343.6
ALFWorld	SPrPO	171.5	400.1	572.9
WebShop	Vanilla	50.1	60.0	110.6
WebShop	SPrPO	58.2	138.7	197.4

C.3 Evaluation Details

General protocol. All evaluations use vLLM (Kwon et al., 2023) inference, and use temperature 0.4 for inference, following GiGPO (Feng et al., 2025). WebShop is evaluated on all 500 goals, while ALFWorld uses the official validation split consisting of 140 ID and 134 OOD gamefiles. All perturbations below are applied only during evaluation to a fixed trained policy.

Channel-wise Gaussian perturbation. We use the same multiplicative channel-wise perturbation as during training. For the FFN activation $h_{\ell c}$ before the down-projection, we apply $\widetilde{h}_{\ell c} = (1 + \xi_{\ell c}) h_{\ell c}$, where $\xi_{\ell c} \sim \mathcal{N}(0, \sigma^2)$. We sweep $\sigma \in \{0.02, 0.05, 0.10, 0.15, 0.20\}$ on both WebShop and ALFWorld.

Channel dropout. We apply a Bernoulli mask to randomly remove FFN channels, independently dropping each channel with probability p. We sweep $p \in \{0.01, 0.02, 0.03, 0.04, 0.05\}$ on both WebShop and ALFWorld.

Structured pruning (Fisher importance). We prune FFN intermediate channels according to a diagonal-Fisher (Molchanov et al., 2019) importance score. For channel c in layer ℓ, we estimate $\widehat{F}_{\ell c} = \mathbb{E}_{\text{calib}}[(h_{\ell c}\,\partial\mathcal{L}/\partial h_{\ell c})^2]$ using a fixed calibration set of 128 agent-interaction sequences collected from a Vanilla-trained policy on the corresponding benchmark. All compared methods share this calibration set, while the Fisher scores are computed separately for each model using its own activations and gradients. Channels are ranked by $\widehat{F}_{\ell c}$, and the lowest-scoring fraction r is removed by masking the corresponding rows of $W_{\text{gate},\ell}$ and $W_{\text{up},\ell}$ and columns of $W_{\text{down},\ell}$. We sweep $r \in \{0.05, 0.10, 0.15, 0.20\}$ on both WebShop and ALFWorld.

Unstructured pruning (SparseGPT). We use SparseGPT (Frantar & Alistarh, 2023) to prune a fraction r of weights in each FFN projection matrix $W_{\text{gate},\ell}$, $W_{\text{up},\ell}$, and $W_{\text{down},\ell}$, while leaving all attention weights dense. SparseGPT is applied sequentially across transformer layers using its Hessian-based weight selection and compensation procedure, with block size 128 and Hessian damping $\lambda = 0.01$. Calibration uses 128 sequences drawn from the same rollout-derived calibration pool used for structured pruning. We sweep $r \in \{0.10, 0.20, 0.30, 0.40\}$ on WebShop and $r \in \{0.10, 0.30, 0.50, 0.70\}$ on ALFWorld.

Quantization (round-to-nearest). We apply round-to-nearest (RTN) quantization to all linear weights in the transformer layers, including both attention and FFN projections. Weights are partitioned into groups of 128, with each group sharing a quantization scale. For a group $\mathcal{G}$ and bit width b, we use $s_{\mathcal{G}} = \max_{w\in\mathcal{G}} |w|/(2^{b-1} - 1)$. Each weight is divided by this scale, rounded to the nearest integer code, and clipped to the representable range. We evaluate INT8, INT6, and INT4 quantization.

D More Experiment Results

Main tables. We report the robustness results across all evaluated perturbation types and strengths. Results are mean $\pm$ standard deviation over three seeds, with boldface denoting the highest mean, including ties.

Table 5: ALFWorld: overall success rate (%).

Perturbation	Strength	Vanilla	Gaussian	SAM	**SPrPO**
Clean	BF16	87.59 ± 0.97	89.05 ± 0.97	86.86 ± 0.73	**91.85** ± 0.42
Gaussian	$\sigma = 0.02$	88.44 ± 1.38	90.02 ± 0.21	86.01 ± 0.76	**92.70** ± 0.36
	$\sigma = 0.05$	87.23 ± 1.67	89.54 ± 1.28	86.74 ± 0.92	**91.12** ± 0.56
	$\sigma = 0.10$	84.79 ± 2.23	85.89 ± 0.56	86.13 ± 0.73	**89.78** ± 0.63
	$\sigma = 0.15$	79.81 ± 0.92	81.51 ± 1.11	81.63 ± 2.48	**86.13** ± 1.90
	$\sigma = 0.20$	64.96 ± 2.55	72.26 ± 0.97	66.79 ± 2.55	**75.91** ± 0.97
Dropout	$p = 0.01$	82.97 ± 2.43	86.50 ± 2.63	85.16 ± 1.65	**88.93** ± 0.21
	$p = 0.02$	79.81 ± 0.84	81.87 ± 3.37	81.51 ± 1.17	**86.13** ± 0.97
	$p = 0.03$	71.53 ± 1.32	73.84 ± 5.59	74.33 ± 3.60	**79.93** ± 1.93
	$p = 0.04$	63.50 ± 1.67	65.82 ± 1.69	63.14 ± 2.55	**73.36** ± 2.19
	$p = 0.05$	54.62 ± 1.65	57.79 ± 1.80	55.60 ± 0.76	**64.60** ± 3.48
Structured pruning	$r = 0.05$	83.82 ± 1.38	83.45 ± 1.11	85.64 ± 1.72	**89.17** ± 0.56
	$r = 0.10$	72.38 ± 2.23	77.25 ± 0.56	72.99 ± 0.36	**83.21** ± 0.73
	$r = 0.15$	59.61 ± 2.01	67.88 ± 0.73	49.76 ± 4.59	**70.44** ± 2.03
	$r = 0.20$	41.73 ± 5.32	23.72 ± 2.28	22.02 ± 1.69	**47.08** ± 7.65
Unstructured pruning	$r = 0.10$	87.96 ± 0.63	88.81 ± 2.48	87.23 ± 0.97	**91.85** ± 1.28
	$r = 0.30$	87.35 ± 0.56	87.59 ± 0.73	86.25 ± 0.56	**91.00** ± 0.56
	$r = 0.50$	84.31 ± 1.59	83.33 ± 0.92	80.90 ± 2.14	**90.39** ± 2.01
	$r = 0.70$	41.12 ± 4.35	32.85 ± 1.93	50.97 ± 1.69	**58.03** ± 2.39
Quantization	INT8	88.44 ± 0.92	88.81 ± 0.84	86.86 ± 0.97	**91.24** ± 0.73
	INT6	86.98 ± 0.84	87.47 ± 0.92	86.25 ± 0.56	**91.24** ± 0.00
	INT4	47.81 ± 3.18	39.17 ± 4.04	**59.61** ± 2.56	48.05 ± 4.02

Table 6: ALFWorld: ID and OOD success rates (%).

		ID				OOD			
Perturbation	Strength	Vanilla	Gaussian	SAM	**SPrPO**	Vanilla	Gaussian	SAM	**SPrPO**
Clean	BF16	89.05 ± 1.09	89.29 ± 1.24	90.48 ± 2.06	**94.05** ± 0.82	86.07 ± 1.55	88.81 ± 1.97	83.08 ± 2.62	**89.55** ± 0.00
Gaussian	$\sigma = 0.02$	89.52 ± 2.70	90.24 ± 0.82	88.33 ± 0.41	**93.33** ± 1.65	87.31 ± 0.75	89.80 ± 0.43	83.58 ± 1.49	**92.04** ± 1.14
	$\sigma = 0.05$	86.90 ± 1.80	90.48 ± 1.49	89.29 ± 1.43	**92.14** ± 1.43	87.56 ± 1.55	88.56 ± 1.14	84.08 ± 2.83	**90.05** ± 0.43
	$\sigma = 0.10$	86.90 ± 2.89	85.95 ± 1.49	88.33 ± 1.80	**90.48** ± 2.30	82.59 ± 3.02	85.82 ± 1.49	83.83 ± 0.43	**89.05** ± 2.28
	$\sigma = 0.15$	77.62 ± 1.65	82.86 ± 2.14	**85.71** ± 3.11	**85.71** ± 2.14	82.09 ± 0.75	80.10 ± 1.88	77.36 ± 2.28	**86.57** ± 4.48
	$\sigma = 0.20$	65.48 ± 2.89	71.67 ± 1.09	67.14 ± 3.11	**76.90** ± 1.65	64.43 ± 3.02	72.89 ± 1.55	66.42 ± 2.59	**74.88** ± 0.43
Dropout	$p = 0.01$	84.76 ± 1.80	87.86 ± 1.89	88.33 ± 1.09	**89.29** ± 1.89	81.09 ± 3.11	85.07 ± 3.88	81.84 ± 3.11	**88.56** ± 1.55
	$p = 0.02$	79.05 ± 0.41	82.62 ± 2.70	83.57 ± 0.71	**87.14** ± 0.71	80.60 ± 1.29	81.09 ± 4.11	79.35 ± 2.62	**85.07** ± 1.49
	$p = 0.03$	73.10 ± 1.49	75.48 ± 5.07	75.24 ± 2.51	**80.24** ± 1.49	69.90 ± 1.14	72.14 ± 8.22	73.38 ± 4.97	**79.60** ± 2.83
	$p = 0.04$	63.81 ± 2.70	68.81 ± 3.38	62.86 ± 1.43	**73.10** ± 4.06	63.18 ± 4.56	62.69 ± 0.75	63.43 ± 3.73	**73.63** ± 1.14
	$p = 0.05$	52.62 ± 1.09	58.33 ± 3.67	55.24 ± 2.30	**65.00** ± 4.34	56.72 ± 3.95	57.21 ± 2.40	55.97 ± 2.24	**64.18** ± 3.95
Structured pruning	$r = 0.05$	86.67 ± 0.82	85.24 ± 2.51	87.86 ± 0.71	**92.86** ± 0.71	80.85 ± 2.40	81.59 ± 4.37	83.33 ± 2.83	**85.32** ± 1.55
	$r = 0.10$	72.38 ± 3.38	77.86 ± 0.71	72.62 ± 1.09	**83.57** ± 0.71	72.39 ± 3.73	76.62 ± 1.72	73.38 ± 1.14	**82.84** ± 1.97
	$r = 0.15$	56.90 ± 3.67	65.95 ± 1.49	48.81 ± 2.51	**66.90** ± 3.22	62.44 ± 3.37	69.90 ± 2.40	50.75 ± 6.84	**74.13** ± 1.14
	$r = 0.20$	36.19 ± 3.93	21.43 ± 2.47	16.43 ± 2.47	**40.71** ± 5.58	47.51 ± 6.77	26.12 ± 3.42	27.86 ± 0.86	**53.73** ± 10.10
Unstructured pruning	$r = 0.10$	88.33 ± 1.09	89.76 ± 2.97	90.48 ± 2.89	**92.38** ± 0.82	87.56 ± 0.43	87.81 ± 3.02	83.83 ± 1.14	**91.29** ± 1.88
	$r = 0.30$	87.14 ± 0.71	87.38 ± 1.49	89.05 ± 0.41	**92.86** ± 1.89	87.56 ± 0.43	87.81 ± 2.83	83.33 ± 1.14	**89.05** ± 0.86
	$r = 0.50$	84.29 ± 2.58	84.76 ± 1.09	84.29 ± 2.58	**91.90** ± 2.18	84.33 ± 0.75	81.84 ± 0.86	77.36 ± 1.72	**88.81** ± 1.97
	$r = 0.70$	41.43 ± 4.68	31.43 ± 1.24	51.90 ± 1.80	**53.81** ± 3.52	40.80 ± 4.80	34.33 ± 3.25	50.00 ± 3.73	**62.44** ± 2.62
Quantization	INT8	89.29 ± 1.24	89.76 ± 1.80	90.24 ± 1.65	**91.90** ± 0.82	87.56 ± 0.86	87.81 ± 1.14	83.33 ± 3.02	**90.55** ± 1.72
	INT6	87.38 ± 0.82	88.81 ± 2.06	89.05 ± 1.09	**91.67** ± 1.49	86.57 ± 1.49	86.07 ± 3.02	83.33 ± 0.43	**90.80** ± 1.55
	INT4	47.62 ± 2.06	42.62 ± 5.46	**57.86** ± 4.95	49.76 ± 6.64	48.01 ± 4.37	35.57 ± 2.62	**61.44** ± 2.28	46.27 ± 1.29

Table 7: WebShop: success rate (%).

Perturbation	Strength	Vanilla	Gaussian	SAM	**SPrPO**
Clean	BF16	57.00 ± 4.39	50.60 ± 3.99	52.27 ± 4.02	**60.00** ± 0.72
Gaussian	$\sigma = 0.02$	57.00 ± 3.82	51.07 ± 3.35	52.13 ± 2.95	**59.73** ± 1.55
	$\sigma = 0.05$	53.47 ± 4.88	48.47 ± 4.80	49.00 ± 2.31	**58.67** ± 1.29
	$\sigma = 0.10$	45.87 ± 6.65	41.53 ± 5.35	40.00 ± 4.13	**53.07** ± 1.29
	$\sigma = 0.15$	32.73 ± 2.95	30.33 ± 7.83	27.27 ± 3.69	**44.33** ± 0.50
	$\sigma = 0.20$	18.73 ± 3.03	17.87 ± 5.12	15.20 ± 2.65	**30.13** ± 1.81
Dropout	$p = 0.01$	44.33 ± 5.83	40.67 ± 6.30	38.67 ± 4.39	**54.87** ± 0.42
	$p = 0.02$	35.20 ± 4.42	31.80 ± 6.93	29.73 ± 5.35	**46.13** ± 1.81
	$p = 0.03$	25.53 ± 5.25	24.07 ± 6.20	20.80 ± 3.82	**38.40** ± 2.80
	$p = 0.04$	18.60 ± 2.95	16.27 ± 4.91	13.87 ± 2.91	**29.47** ± 2.58
	$p = 0.05$	12.20 ± 3.30	11.33 ± 2.60	8.87 ± 2.14	**20.87** ± 2.00
Structured pruning	$r = 0.05$	49.93 ± 8.60	47.80 ± 3.17	45.33 ± 2.80	**55.80** ± 3.52
	$r = 0.10$	30.13 ± 7.80	27.40 ± 10.51	20.20 ± 5.37	**43.87** ± 4.15
	$r = 0.15$	6.33 ± 4.05	18.67 ± 7.20	6.60 ± 2.75	**23.27** ± 2.58
	$r = 0.20$	2.27 ± 1.86	0.27 ± 0.12	0.60 ± 0.87	**12.40** ± 3.65
Unstructured pruning	$r = 0.10$	57.07 ± 3.90	50.00 ± 3.30	53.47 ± 3.93	**60.20** ± 0.40
	$r = 0.20$	56.33 ± 3.44	47.67 ± 2.50	50.80 ± 3.65	**60.00** ± 1.06
	$r = 0.30$	51.53 ± 7.06	46.13 ± 6.52	43.53 ± 5.49	**59.53** ± 1.51
	$r = 0.40$	42.00 ± 8.31	34.33 ± 9.00	29.53 ± 5.74	**50.20** ± 2.99
Quantization	INT8	56.67 ± 4.67	50.47 ± 3.88	52.27 ± 4.05	**60.13** ± 0.61
	INT6	45.20 ± 9.18	39.80 ± 9.90	34.93 ± 4.54	**55.67** ± 1.60
	INT4	9.60 ± 2.31	**9.87** ± 5.46	2.20 ± 0.35	8.20 ± 2.42

Ablation and hyperparameter results. We report WebShop ablations and hyperparameter sweeps below. Vanilla ($\rho = 0$) and full SPrPO ($\rho = 0.2$, $\alpha = 1$) reuse results from Table 1; other configurations use separate ablation evaluation batches.

Table 8: Hyperparameter study. Success rate (%).

Value	Clean	Gaussian $\sigma = .20$	Dropout $p = .05$	Struct. $r = .20$	Unstruct. $r = .40$	INT6
ρ sweep ($\alpha = 1$)						
0	57.00 ± 4.39	18.73 ± 3.03	12.20 ± 3.30	2.27 ± 1.86	42.00 ± 8.31	45.20 ± 9.18
0.1	55.33 ± 5.16	17.67 ± 5.52	10.73 ± 2.93	4.27 ± 4.45	40.80 ± 9.71	39.53 ± 9.98
0.2	**60.00** ± 0.72	**30.13** ± 1.81	**20.87** ± 2.00	**12.40** ± 3.65	**50.20** ± 2.99	**55.67** ± 1.60
0.3	48.27 ± 2.50	25.93 ± 2.02	17.00 ± 1.97	7.60 ± 2.62	40.47 ± 3.56	39.27 ± 4.75
α sweep ($\rho = 0.2$)						
0	48.80 ± 0.80	27.07 ± 3.72	18.93 ± 3.42	11.07 ± 6.92	47.07 ± 3.11	48.27 ± 3.06
0.5	52.87 ± 0.61	24.40 ± 5.10	16.00 ± 5.39	7.07 ± 4.47	38.00 ± 8.01	47.87 ± 3.72
1	**60.00** ± 0.72	**30.13** ± 1.81	**20.87** ± 2.00	**12.40** ± 3.65	**50.20** ± 2.99	**55.67** ± 1.60
2	57.40 ± 2.09	26.80 ± 4.70	19.87 ± 4.36	8.93 ± 6.85	46.60 ± 4.85	52.40 ± 6.13

Table 9: Component ablations on WebShop. Success rate (%).

Training variant	Clean	Gaussian $\sigma = .20$	Dropout $p = .05$	Struct. $r = .20$	Unstruct. $r = .40$	INT6
w/o perturbation ($\rho = 0$)	57.00 ± 4.39	18.73 ± 3.03	12.20 ± 3.30	2.27 ± 1.86	42.00 ± 8.31	45.20 ± 9.18
w/o sensitivity ($\alpha = 0$)	48.80 ± 0.80	27.07 ± 3.72	18.93 ± 3.42	11.07 ± 6.92	47.07 ± 3.11	48.27 ± 3.06
w/o scheduling	52.13 ± 0.42	26.40 ± 4.70	19.33 ± 5.32	3.60 ± 2.88	45.00 ± 4.20	49.73 ± 2.42
Full SPrPO	$\mathbf{60.00} \pm 0.72$	$\mathbf{30.13} \pm 1.81$	$\mathbf{20.87} \pm 2.00$	$\mathbf{12.40} \pm 3.65$	$\mathbf{50.20} \pm 2.99$	$\mathbf{55.67} \pm 1.60$

E QUALITATIVE EXAMPLES UNDER POLICY PERTURBATIONS

In this section, we provide qualitative examples comparing Vanilla and SPrPO on ALFWorld and WebShop under different policy perturbations.

Case 1 ALFWorld: malformed action tags

Perturbation: Gaussian noise $\sigma = 0.2$ | ID

Task. put two soapbar in garbagecan.
Without perturbation. Both succeed: Vanilla 8 steps; SPrPO 9 steps.

Shared decision context — before Step 2
Both first go to countertop 1, where the observation lists "a soapbar 4, a soapbar 1". Both unperturbed controls also receive this identical complete input.

Vanilla — FAIL

Step 2
`<take soapbar 1 from countertop 1>`
Feedback: "Nothing happens."

Required action tags are missing.

Step 3
`<action>take soapbar 1 from countertop 1</action>`
Feedback: "You pick up the soapbar 1 from the countertop 1."

Step 4
`<move soapbar 1 to garbagecan 1>`
Feedback: "Nothing happens."

Step 5 omitted: goes to cabinet 1.

Step 6–7 (same output twice)
`<open cabinet 1>`

Step 8
`<examine cabinet 1>`

Step 9
`<go to garbagecan 1>`

Outcome: failure.
6 of the 9 recorded outputs have malformed tags.

SPrPO — SUCCESS

Step 2
`<action>take soapbar 4 from countertop 1</action>`
Feedback: "You pick up the soapbar 4 from the countertop 1."

Step 3 omitted: goes to garbagecan 1.

Step 4
`<action>move soapbar 4 to garbagecan 1</action>`
Feedback: "You move the soapbar 4 to the garbagecan 1."

Steps 5–6 omitted: visits toilet 1, then returns to countertop 1.

Step 7
`<action>take soapbar 1 from countertop 1</action>`
Feedback: "You pick up the soapbar 1 from the countertop 1."

Step 8 omitted: goes to garbagecan 1.

Step 9
`<action>move soapbar 1 to garbagecan 1</action>`

Outcome: success in 9 steps.
All 9 outputs retain valid action tags; both soapbars reach the garbagecan.

What this shows. Vanilla briefly recovers at Step 3 but repeatedly loses the required format. Its first failed command names an admissible action, accepted in a parser-only check after restoring the tags. This identifies an interface failure, without establishing that a tag repair would rescue the whole episode. SPrPO preserves the format and completes the task.

Case 2 ALFWorld: undoing subgoal progress

Perturbation: Dropout $p = 0.01$ | OOD

Task. find two cd and put them in safe.
Without perturbation. Both succeed: Vanilla 18 steps; SPrPO 20 steps.

Shared decision context — before Step 3
Both visit desk 1 and take cd 1 in Steps 1–2. Current observation: "You pick up the cd 1 from the desk 1." The task still requires placing *both* CDs in the safe.

Vanilla — FAIL

Step 3
`go to shelf 1`

Step 4
`move cd 1 to shelf 1`
Feedback: "You move the cd 1 to the shelf 1."

Steps 5–8 omitted: takes cd 2 from desk 2, reaches and opens the safe.

Step 9
`move cd 2 to safe 1`
Feedback: "You move the cd 2 to the safe 1."

Steps 10–18 omitted: retrieves cd 1, leaves it on shelf 5, returns to the safe.

Step 19
`take cd 2 from safe 1`

Step 20
`move cd 2 to safe 1`

Retrieves and replaces an already delivered CD.

Steps 21–37 omitted: further relocations and repeated visits.

Step 38
`move cd 1 to shelf 4`
Feedback: "You move the cd 1 to the shelf 4."

Steps 39–49 omitted: removes cd 2 again, leaves it on shelf 6, then retrieves it.

Step 50
`move cd 2 to safe 1`

Outcome: failure at the 50-step limit.
cd 1 remains on shelf 4; only cd 2 is in the safe.

SPrPO — SUCCESS

Step 3
`go to safe 1`

Step 4
`open safe 1`

Step 5
`move cd 1 to safe 1`
Feedback: "You move the cd 1 to the safe 1."

First delivery is complete.

Step 6
`go to desk 2`

Step 7
`take cd 2 from desk 2`
Feedback: "You pick up the cd 2 from the desk 2."

Step 8
`go to safe 1`

Observation before Step 9:
"The safe 1 is open. In it, you see a cd 1."

Step 9
`move cd 2 to safe 1`

Outcome: success in 9 steps.
Both CDs are in the safe. The first delivery is preserved while the second is completed.

What this shows. The failure is a sequence of executable but unproductive actions: Vanilla moves cd 1 among shelves and repeatedly removes cd 2 from the target container. Temporary storage alone does not establish failure; the later relocations and final state show that the two-object goal remains incomplete. SPrPO completes the two deliveries in sequence without undoing the first.

Case 3 WebShop: invalid click targets after pruning

Perturbation: Structured pruning, ratio 0.20 | goal 309

Task. Find me eco friendly, heavy duty, assembly required, easy clean file cabinets with size: 23.6"w x 15.7"d x 29.3"h, and price lower than 200.00 dollars
Without perturbation. Both succeed in 4 steps (task score 1.0).

Shared decision context — before Step 5
After different search queries (Step 1 omitted), both open product `B08MFGNXW3` at Step 2 and select `23.6"w x 15.7"d x 29.3"h` at Steps 3 and 4. Thus, *both have already repeated the size selection once*. Page excerpts: "DEVAISE 2-Drawer Wood Lateral File Cabinet with Lock for Office Home, Black"; "Price: $169.99"; "Buy Now".

Vanilla — FAIL

Step 5
`click[23.6"w x 15.7"d x 29.3"h]`
Selects the same size a third time.

Step 6
`click[back to search]`

Step 7
`search[eco friendly, heavy duty, assembly required file cabinets]`
A results page lists product IDs and navigation links.

Step 8
`click[eco friendly, heavy duty, assembly required file cabinets]`
The query text is not a clickable target. The results page does not change.

Steps 9–10 omitted: returns to search and issues a shorter query.

Step 11–15 (same action five times)
`click[eco friendly, heavy duty file cabinets]`
Again clicks query text that is absent from the available targets.

Outcome: failure at the 15-step limit.
No purchase; task score 0.0.

SPrPO — SUCCESS

Step 5
`click[Buy Now]`

Outcome: success in 5 steps.
Task score 1.0.
Purchases the already selected product with the requested size. The episode ends here.

Decision contrast.
At the identical Step-5 input, SPrPO proceeds to purchase. Vanilla first repeats the option selection, then abandons the product and repeatedly clicks search-query text.

What this shows. Vanilla reaches the product page and selects the requested size, but fails to finish the transaction. Its later commands use a valid `click[...]` structure with targets that the page does not provide. SPrPO also repeats the size selection once, but then buys.

Case 4 WebShop: option repetition without purchase

Perturbation: INT6 weight quantization (RTN) | goal 295

Task. Find me loose fit, day comfort, hand wash women's tops, tees & blouses with short sleeve, polyester spandex for teen girls with color: a6-black, and size: xx-large, and price lower than 40.00 dollars

Without perturbation. Both succeed in 5 steps (task score 1.0).

Shared decision context — before Step 5

After their search queries (Step 1 omitted), both execute the same action text in Steps 2–4:

```
click[B09NPML43M]
click[a6-black]
click[xx-large]
```

Both requested options have been selected. The product page still offers "xx-large" and "Buy Now", and lists "Price: $1.01 to $1.74".

Vanilla — FAIL

Step 5

```
click[xx-large]
```

Repeats the selected size instead of buying.

Step 6–15 (same action ten times)

```
click[xx-large]
```

The product-page observation remains unchanged throughout this loop.

Outcome: failure at the 15-step limit.
12 consecutive size clicks in total (Steps 4–15); no purchase; task score 0.0.

SPrPO — SUCCESS

Step 5

```
click[Buy Now]
```

Outcome: success in 5 steps.
Task score 1.0.

Purchases after selecting the requested color and size. The episode ends here.

Decision contrast.
The same Step-5 input leads to another size click for Vanilla and the terminal purchase action for SPrPO.

What this shows. Vanilla reaches the product and selects both requested options, but then repeatedly clicks the already selected size until the step limit. Its actions remain well formed and refer to an option on the page, yet never complete the transaction. SPrPO proceeds to purchase from the same decision input.