

BEYOND WORST-CASE: EFFICIENT ROBUST RL VIA IN-CONTEXT GENERALIZATION

Anonymous authors

Paper under double-blind review

ABSTRACT

Robustness and generalization are central challenges in reinforcement learning (RL). Classical robust RL handles perturbations with worst-case minimax optimization, which is hard to solve in practice and often yields pessimistic policies with suboptimal performance during deployment. In contrast, in-context RL (ICRL), where pretrained transformers adapt to new tasks without parameter updates, is designed for generalization. Rather than treating robustness and generalization as orthogonal, we demonstrate an interesting link that robustness can emerge as a consequence of generalization in ICRL, and that generalization can be systematically leveraged to improve robustness. Specifically, we apply ICRL models for robust RL and observe that even without explicit robust training, in-context models perform strongly on robustness benchmarks. Motivated by this, we investigate the robustness of ICRL models and identify significant performance degrade under disturbances. To address this limitation, building on the insight that we can transform robustness challenges to generalization problems, we propose in-context adversarial pretraining, which augments pretraining tasks with environment perturbations to expand diversity without solving the minimax game. We also introduce an adaptive rollout variant that uses the pretrained transformer to generate high-value trajectories online, improving coverage and sample efficiency. Across navigation and continuous-control benchmarks, these strategies consistently improve robustness and nominal return, offering a scalable path to robust performance without worst-case optimization or significant performance sacrifice.

1 INTRODUCTION

Robustness and generalization are fundamental desiderata in evaluating reinforcement learning (RL) agents. Classical approaches to robust RL often rely on worst-case minimax optimization (Moos et al., 2022; Pinto et al., 2017), which is both computationally challenging and prone to producing overly conservative policies (Dong et al., 2025b), leading to suboptimal performance at deployment. By contrast, in-context RL (ICRL) has recently attracted considerable attention for its ability to generalize to novel tasks. In ICRL, we pretrain transformer models (TMs) on a diverse suite of RL tasks and then deploy them to novel ones. Conditioned on a *context dataset* of trajectories collected either by unknown behavior policies (offline) or by the TMs themselves (online), the pretrained TMs adapt within only a few trajectories, often outperforming RL policies trained from scratch with thousands of trajectories (Laskin et al., 2023; Lee et al., 2024). Notably, ICRL achieves this *without parameter updates*, enabling deployment that is both convenient and cost-efficient. While robustness and generalization appear orthogonal, this work demonstrates and investigates an interesting link between them that (i) *robustness can emerge as a consequence of generalization in in-context RL*, and that (ii) *generalization can be systematically leveraged to improve robustness*.

Generalization for Robustness. Motivated by the **deployment parity** that both robust RL policies and ICRL models require no parameter updates at test time, we repurpose ICRL for robust RL. In this view, a pretrained ICRL model acts as a drop-in robust policy that adapts from contextual trajectories at deployment, incurring no additional overhead. In Figure 1, we compare the *Decision-Pretrained Transformer* (DPT) (Lee et al., 2024), a representative ICRL framework, against robust RL baselines. Using *Quadruped* and *Cheetah* (Tunyasuvunakool et al., 2020), two classical robust-RL testbeds, we evaluate two canonical robust RL settings: (i) **multi-domain**, where agents train on an ensemble of environments (e.g., simulations with varying mass–friction pairs) and (ii)

054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
100
101
102
103
104
105
106
107

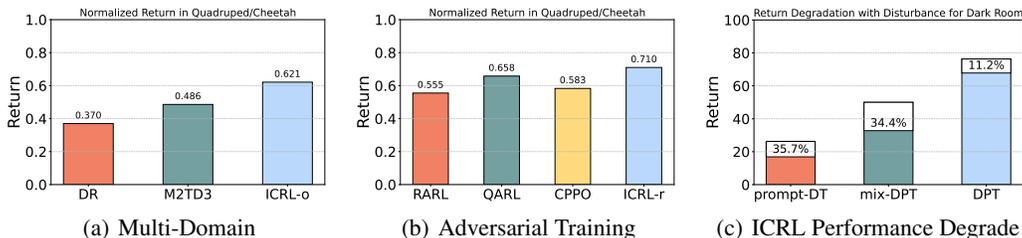


Figure 1: Comparison of ICRL to robust-RL methods and the Robustness of ICRL Methods. **(a, b)** Normalized return under two classical robust RL testbeds: **Quadruped and Cheetah**. **(c)** Return degradation of in-context RL methods under action perturbation ($p = 0.2$) in the Dark Room environment. Bars indicate nominal performance; hatched regions show return loss due to disturbance.

adversarial training, where agents learn in zero-sum two-player games within a single environment. For multi-domain, we compare against strong baselines including domain randomization (Tobin et al., 2017) and M2TD3 (Tanabe et al., 2022); for adversarial training, we use RARL (Pinto et al., 2017), QARL (Reddi et al., 2024), and CVaR-PPO (CPPO) (Ying et al., 2022) as counterparts. We apply ICRL directly to the multi-domain setting, denoted **ICRL-original**. For adversarial training, we cast each adversary as a distinct task, yielding **ICRL-robust**. As can be observed in Figures 1(a) and 1(b), across both classical robust RL settings, ICRL methods outperform robust-RL baselines, even though the ICRL models are pretrained *without* any robustness objective. Importantly, ICRL and robust RL methods are trained under *identical* environment resources (i.e., access to multi-domains and an environment with adjustable adversaries), only differing in their models and training algorithms. Hence, applying ICRL to robust RL yields *efficient robustness*: it avoids both the performance sacrifices typical of robustness-oriented methods and the computational challenge of max-min optimization.

Robustness of Generalization. Motivated by this observation, we examine the robustness properties of ICRL models more closely. We find that, despite of ICRL’s strong performances for robust RL settings, those scores only reflect degraded returns under disturbance. In Figure 1(c), we present experiments in the Dark Room environment (Laskin et al., 2023) to investigate the performance of pretrained TMs under deployment disturbances. Our results show that their performance degrades under even mild disturbances, specifically with action perturbation applied at probability $p=0.2$. Although the magnitude of degradation varies across methods (i.e., DPT, prompt-DT (Xu et al., 2022), mix-DPT (Wang et al., 2024)), the consistent downward trend reveals a common vulnerability: *While ICRL models are pretrained to adapt to new tasks or environmental changes, they still exhibit limited robustness to perturbations.*

This observation motivates us to design an ICRL framework with *robust performance* under disturbance while maintaining *strong generalization* to new tasks. To address this challenge, we draw inspiration from robust RL, which finds policies with the best worst-case performance among an uncertainty set through solving a max-min problem. In the sequel, we will refer to these policies obtained through max-min optimizations as *robust policies*. Notably, within the supervised pretraining framework for ICRL (Lee et al., 2024), the worst-case optimization approach from robust RL can be directly extended to ICRL by simply modifying the action labels used during pretraining. See Figure 2(b) and Section 3 for details. In this case, we pretrain TMs to *infer robust actions* that would be taken by robust policies in new RL tasks. However, this approach has several known limitations related to the worst-case optimization, including its sensitivity on the uncertainty set and the challenges of solving the max-min optimization. See Section 3.2 for detailed discussions.

Efficient Robustness Through Generalization. Our insight for addressing these challenges is that we can use the in-context learning capability of TMs to *transform the robust optimization (i.e., max-min) problems into generalization problems*. Instead of pretraining the TMs to infer a robust policy for a new RL task, we propose to *directly* pretrain them to *generalize* across varying disturbances and adversaries, moving beyond worst-case optimizations for robustness. We term this proposed approach *In-context Adversarial Generalization (ICAG)*. Specifically, ICAG augments the pretraining dataset to include perturbed environments where identifying the optimal policies can be parallelized and considerably easier than finding a robust policy through max-min optimization.

Subsequently, pretraining on the augmented dataset with trajectories from the perturbed environments facilitates TMs’ ability to generalize to unseen disturbances and adversary attacks. Hence, ICAG achieves efficient (ICRL) robustness without expensive and challenging worst-case optimization. See Figure 2(c) for a conceptual demonstration. Furthermore, we observe that ICAG also improves ICRL performance when deployed to new environments *without* disturbances. This is because ICAG also functions as a data augmentation method, where the generated disturbances effectively increase the diversity of pretraining tasks to help generalization.

Contributions. We summarize our contributions below.

- C1.** Building on an observed linkage between robustness and generalization, we propose to transform the robustness problems to generalization problems.
- C2.** Motivated by this insight, we apply ICRL methods for robust RL and observe that ICRL are strong robust RL agents, despite being pretrained without any robust RL objectives.
- C3.** We further realize that the already competitive performance of ICRL models can be further enhanced by increasing the robustness of ICRL models. To this end, we propose the ICAG algorithm. Unlike conventional robust RL, by transforming the robustness problem into the generalization problem, we can enhance the robustness of ICRL models without any worst-case (max-min) optimizations, which are inefficient and difficult to solve.
- C4.** Noticing that the execution of ICAG can be time-consuming and expensive, we propose ICAA, an approximation of ICAG that bootstraps on the existing generalization capability of ICRL models to efficiently collect new pretraining data, thereby increasing the scalability.

Throughout extensive experiments on sparse-reward navigation tasks (Laskin et al., 2023) and two complex continuous control tasks (Meta-World and MuJoCo with 11 problems (Todorov et al., 2012)), our ICAG and ICAA demonstrate superior performance, particularly when under agent disturbances during deployment.

2 PRELIMINARY

Markov Decision Process (MDP). Sequential decision-making tasks are modeled as MDPs (Kaelbling et al., 1996; Busoniu et al., 2008). An MDP τ is defined by the tuple $(\mathcal{S}, \mathcal{A}, P_\tau, R_\tau, \gamma, \rho_\tau)$, where \mathcal{S} is the set of states, \mathcal{A} is the set of actions, $P_\tau : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ is the state transition function, $R_\tau : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, $\gamma \in (0, 1)$ is the discount factor, and $\rho_\tau \in \Delta(\mathcal{S})$ is the initial state distribution. At each time step h , an agent selects an action $a_h \in \mathcal{A}$, receives a reward $r_h = R_\tau(s_h, a_h)$, and transitions to the next state s_{h+1} according to $P_\tau(s_h, a_h)$. A policy $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ maps the current state to an action distribution. The agent’s goal is to learn the optimal policy π_τ^* that maximizes the expected cumulative reward $G_\tau(\pi) = \mathbb{E}[\sum_{h=1}^{\infty} \gamma^{h-1} r_h \mid \pi, \tau]$.

Supervised Pretraining. Our approach builds on the DPT architecture (Lee et al., 2024), a supervised pretraining method designed to equip transformer models with ICRL capabilities. In DPT, a set of tasks $\{\tau^i\}_{i=1}^m$ is drawn from a task distribution p_τ . Each task $\tau^i \in \mathcal{M}$ represents an instance of an MDP where \mathcal{M} is the space of all tasks of interest, and for each task, a context dataset D^i is generated from interactions between a behavioral policy and τ^i , i.e., $D^i = \{(s_h^i, a_h^i, s_{h+1}^i, r_h^i)\}_h$, where a_h^i is selected by the behavioral policy. For each task τ^i , a query state $s_{\text{query}}^i \in \mathcal{S}$ is chosen, and the optimal action a_i^* is sampled from $\pi_{\tau^i}^*(s_{\text{query}}^i)$, where $\pi_{\tau^i}^*$ is the optimal policy for τ^i . The complete pretraining dataset is then denoted as $\mathcal{D}_{pre} = \{D^i, s_{\text{query}}^i, a_i^*\}_{i=1}^m$. Let T_θ represent a causal GPT-2 transformer with parameters θ (Radford et al., 2019). The pretraining objective of DPT is formulated as follows:

$$\min_{\theta} \frac{1}{m} \sum_{i=1}^m -\log T_\theta(a_i^* | s_{\text{query}}^i, D^i). \quad (1)$$

ICRL Deployment. The pretrained autoregressive TM T_θ can be deployed as an agent in both **offline** and **online** settings. During deployment, an unseen task τ is sampled from the task distribution p_τ . In offline deployment, a dataset D_{off} is first collected from τ , typically using trajectories generated by a random policy. Once available, DPT selects actions based on the policy $T_\theta(\cdot | s_h, D_{\text{off}})$ after observing the state s_h at time step h . For online deployment, DPT begins with an empty dataset D_{on} . At each episode, DPT follows $T_\theta(\cdot | s_h, D_{\text{on}})$ to collect a trajectory $\xi = \{s_1, a_1, r_1, \dots, s_H, a_H, r_H\}$, which

is appended to D_{on} . This process repeats for a predefined number of episodes. The pseudocode for this process is provided in Algorithm 3 in Appendix H.

3 EXTENDING ROBUST RL FOR ROBUST ICRL

To improve the robustness of ICRL methods, a natural starting point is to adapt existing robust RL techniques developed for standard RL settings to the ICRL context. In Section 3.1, we briefly review adversarial training, a widely used technique for enhancing policy robustness in standard RL. Then, in Section 3.2, we explore how adversarial training can be directly extended to ICRL within the supervised pretraining framework and highlight its inherent limitations in this setting. These limitations, in turn, motivate our proposed frameworks in Section 4, which leverage the in-context learning capabilities of transformer models to overcome these challenges more effectively.

3.1 ROBUST RL FROM ADVERSARIAL TRAINING

While the objective of standard RL is to find a policy π_τ that performs well for a specific target MDP τ , robust RL aims to ensure performance even when the deployment MDP τ' differs from the pre-specified target τ (Moos et al., 2022). Adversarial training is one of the most effective robust RL methods, which proposes to learn a policy robust to adversarial attacks and disturbances (Pinto et al., 2017). Adversarial training has shown great success in improving the policy robustness to both (i) environment mis-specification and (ii) external disturbances. It can be formulated as a **Markov Game**, defined by a tuple of 6 elements $\tau^G = (\mathcal{S}, \mathcal{A}, \mathcal{A}^a, P_\tau^G, R_\tau^G, \gamma, \rho_\tau)$; here, \mathcal{S} , γ , and ρ_τ are defined as in the MDP formulation, representing the set of states, the discount factor, and the initial state distribution, respectively; \mathcal{A} and \mathcal{A}^a are respectively the sets of actions that the agent (protagonist) and the adversary can take; $P_\tau^G : \mathcal{S} \times \mathcal{A} \times \mathcal{A}^a \rightarrow \Delta(\mathcal{S})$ is the transition function that describes the distribution of the next state given the current state and actions taken by the agent and the adversary; $R_\tau^G : \mathcal{S} \times \mathcal{A} \times \mathcal{A}^a \rightarrow \mathbb{R}$ is the reward function for the agent. Consider an adversary $\phi : \mathcal{S} \rightarrow \Delta(\mathcal{A}^a)$. We use $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ to denote the agent policy to learn. Let $s_h \in \mathcal{S}$ be the state of the environment, and let $a_h \in \mathcal{A}$ (respectively $a_h^a \in \mathcal{A}^a$) denote the action of the agent (respectively the adversary) at time step h . We use

$$\mathcal{R}_\tau(\pi, \phi) := \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t R_\tau^G(s_t, a_t, a_t^a) | s_0 \sim \rho_\tau \right] \quad (2)$$

where $a_h \sim \pi(s_h)$, $a_h^a \sim \phi(s_h)$ represents the cumulative discounted reward that the agent π can receive under the disturbance of the adversary following policy ϕ . For a specific target MDP τ , adversarial training solves for a robust policy

$$\pi_\tau^{rb} \in \arg \max_{\pi} \min_{\phi \in \Phi(\tau)} \mathcal{R}_\tau(\pi, \phi), \quad (3)$$

where $\Phi(\tau)$ is a pre-defined adversary space for τ . In this approach, the RL agent π optimizes the worst-case performance among all the adversarial disturbances from Φ .

3.2 IN-CONTEXT ROBUST REINFORCEMENT LEARNING

Although primarily designed for standard RL scenarios, the adversarial training approach can be directly adapted to ICRL within the supervised pretraining framework. We term this approach *In-context Robust RL* (IC2RL), which we illustrate in Figure 2(b) and elaborate next.

Specifically, given the pretraining tasks $\{\tau^i\}_{i=1}^m$, we can construct a dataset for supervised pretraining following two steps. In the first step, we solve for a robust policy $\pi_{\tau^i}^{rb}$ for each task τ^i , defined as

$$\pi_{\tau^i}^{rb} \in \arg \max_{\pi} \min_{\phi \in \Phi(\tau^i)} \mathcal{R}_{\tau^i}(\pi, \phi), \quad (4)$$

where $\Phi(\tau^i)$ is the adversary set for task τ^i . The second step mirrors the data collection process of supervised pretraining in Section 2, except that the sampled query state $s_{\text{query}}^i \in \mathcal{S}$ is annotated with a *robust action label* a_i^{rb} drawn from the robust policy $\pi_{\tau^i}^{rb}(s_{\text{query}}^i)$ rather than the optimal policy $\pi_{\tau^i}^*$ used in standard supervised pretraining. Thus, the complete pretraining dataset for IC2RL is $D_{\text{pre}}^{rb} = \{D^i, s_{\text{query}}^i, a_i^{rb}\}_{i=1}^m$, and we can pretrain a causal transformer with the same pretraining

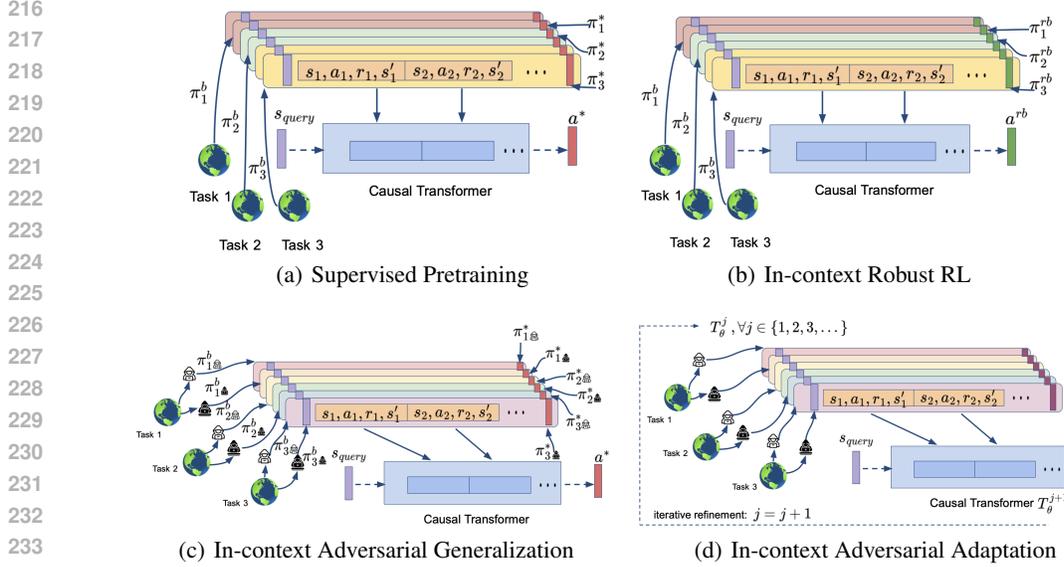


Figure 2: Overview of the supervised pretraining framework for ICRL across multiple tasks, where each task corresponds to an MDP instance τ^i , with $\tau^i \in \{1, 2, 3\}$ as illustrated. **(a)** A TM is pretrained to predict the optimal action across RL tasks (Lee et al., 2024), using optimal action labels a^* from expert policies $\pi_{\tau^i}^*$, given query states s_{query} and offline trajectories collected from different RL tasks by distinct behavioral policies $\pi_{\tau^i}^b$. **(b)** To enhance policy robustness, a robust action label a^{rb} is sampled from the robust policy $\pi_{\tau^i}^{rb}$ rather than $\pi_{\tau^i}^*$. **(c)** To approximate robust learning without explicit max-min optimization, each task τ^i is augmented with K adversarial variations $\{\phi^{i,k}\}_{k=1}^K$ (e.g., black and white hackers), and the optimal action labels are derived from $\pi_{\tau^i, \phi}^*$, representing the optimal policy under disturbance ϕ . **(d)** Alternatively, high-quality action labels \hat{a} and trajectories are generated for K task variants via recursive online adaptation using pretrained models T_{θ}^j , for $j \in \{1, 2, 3, \dots\}$.

objective as in Equation 1. In this ICRL approach, *the TMs are pretrained to infer an action which would be taken by a robust RL policy*.

However, as a straightforward extension of the regular robust RL approaches, this approach inherits a couple of long-standing limitations within robust RL and induces considerable computation cost. **First**, the max-min problem in Equation 4 poses a challenging optimization problem, especially in the current era of deep RL, where such problems are often non-convex and non-concave. Solving this problem requires substantial computational effort to approximate robust RL for a single task. While this computational burden may be manageable in standard robust RL settings with only one task of interest, the (often vast) number of RL tasks involved in ICRL pretraining renders the required computational cost prohibitively high. **Second**, the success of the max-min approach in robust RL relies on effective construction of the adversary parameter set. If not properly constructed, an adversary set which is too broad leads to over-pessimistic policies due to the worst-case optimization, and, on the other hand, an adversary set which is too limited leads to policies with insufficient robustness to disturbances and environment mis-specification (Dong et al., 2025b).

4 IN-CONTEXT ADVERSARIAL GENERALIZATION AND ADAPTATION

In the light of these concerns, in Section 4.1, we first build on our insights about the linkage between robustness and generalization to propose the *In-context Adversarial Generalization (ICAG)* algorithm, which effectively utilizes the generalization ability of ICRL models to enhance their robustness without worst-case optimization. Despite its strong performance across extensive benchmarks (see Section 5), ICAG can incur excessive computation cost and runtime. To address this challenge, in Section 4.2, we further propose *In-context Adversarial Adaptation (ICAA)*, an efficient implementa-

270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323

Algorithm 1 In-Context Adversarial Generalization

```

1: Input: Causal TM  $T_\theta$ ; number
of pretraining tasks  $m$ ; number of
variation environments per task  $K$ 

2: // Construction of
Pretraining Dataset
3: Initialize an empty pretraining
dataset  $\mathcal{D}_{pre}$ 
4: for  $i \in \{1, \dots, m\}$  do
5:   Sample a task  $\tau^i \in \mathcal{M}$ ;
6:   for  $k \in \{1, \dots, K\}$  do
7:     Sample a disturbance  $\phi^{i,k} \in$ 
 $\Phi(\tau^i)$ ;
8:     Collect a context dataset
 $D^{i,k}$  from  $(\tau^i, \phi^{i,k})$ ;
9:     Sample a query state  $s_{query}^{i,k}$ ;
10:    Train  $\pi_{\tau^i, \phi^{i,k}}^*$  follow-
ing Equation 5;
11:    Sample action label  $a_{i,k}^* \sim$ 
 $\pi_{\tau^i, \phi^{i,k}}^*(s_{query}^{i,k})$ ;
12:    Append  $(D^{i,k}, s_{query}^{i,k}, a_{i,k}^*)$ 
into  $\mathcal{D}_{pre}$ ;
13:   end for
14: end for
15: // Supervised Pretraining
Pretrain the ICRL model using
 $\mathcal{D}_{pre}$  and the objective function
in Equation 1.

```

Algorithm 2 In-Context Adversarial Adaptation

```

1: Input: number of pretraining tasks  $m$ ; number of varia-
tion environments per task  $K$ ; initial pretraining dataset
 $\mathcal{D}^0$ 
2: // Initial Supervised Pretraining
3: Pretrain a causal transformer with  $\mathcal{D}^0$  to have  $T_\theta^0$ .
4: Set  $\mathcal{D}_{pre} = \mathcal{D}^0$ .
5: for  $j \in \{0, \dots, J\}$  do
6:   // Collecting New Pretraining Data
7:   Initialize an empty dataset  $\mathcal{D}^{j+1}$ .
8:   for  $i \in \{1, \dots, m\}$  do
9:     Sample a task  $\tau^i \in \mathcal{M}$ ;
10:    for  $k \in \{1, \dots, K\}$  do
11:      Sample a disturbance  $\phi^{i,k} \in \Phi(\tau^i)$ ;
12:      Deploy  $T_\theta^j$  for each variation environ-
ment  $(\tau^i, \phi^{i,k})$  (following Algorithm 3)
with  $N + 1$  trials to have trajectories
 $\xi_n = \{(s_h, a_h, s_{h+1}, r_h)\}_h, n \in \{0, \dots, N\}$ ;
13:      Use the first  $\underline{N} + 1$  trajectories as context
datasets  $D_n = \xi_n, n \in \{0, \dots, \underline{N}\}$ ;
14:      Sample query state-action label pairs from
the remaining trajectories:  $\{(s_{query}^n, a^n)\}_{n=0}^{\underline{N}} \sim$ 
 $\bigcup_{n=\underline{N}}^N \{(s_h, a_h) \in \xi_n\}_{h=0}^{H-1}$ 
15:      Append  $\{D_n, s_{query}^n, a^n\}_{n=0}^{\underline{N}}$  into  $\mathcal{D}^{j+1}$ ;
16:    end for
17:   end for
18:   // Supervised Fine-Tuning
19:    $\mathcal{D}_{pre} = \mathcal{D}_{pre} \cup \mathcal{D}^{j+1}$ .
20:   Pretrain TM  $T_\theta^j$  with  $\mathcal{D}_{pre}$  to obtain  $T_\theta^{j+1}$ .
21: end for

```

tion of ICAG that bootstraps on the existing generalization capability of ICRL models to efficiently collect new pretraining data, thereby increasing the scalability of ICAG.

4.1 IN-CONTEXT ADVERSARIAL GENERALIZATION

Motivational Insight. The proposed framework is motivated by our insight that worst-case optimization is *not necessary* when the model can generalize in-context to various disturbances. In particular, instead of pretraining the TMs to infer actions taken by the robust policies that solve max-min problems, we propose to pretrain TMs to *directly infer the optimal actions to take under various disturbances*. Compared to regular ICRL, the proposed approach has two axes of generalization: one for generalizing across the environments and the other for generalizing across the disturbances. Our proposed framework ICAG achieves generalization on these two axes. Notably, ICAG addresses the aforementioned challenges by entirely circumventing the max-min optimization.

Algorithm. Recall that \mathcal{M} is a set of MDPs representing the space of environments (RL tasks) of interest for ICRL. Our goal is to pretrain TMs to generalize across the following *adversary-augmented* task space

$$\mathcal{M}_v := \{(\tau, \phi) : \tau \in \mathcal{M}; \phi \in \Phi(\tau)\},$$

where $\Phi(\tau)$ is defined as in Equation 4, representing a pre-specified set of adversaries for τ . Here, each pair $(\tau, \phi) \in \mathcal{M}_v$ represents an environment τ along with an associated disturbance policy π_ϕ . When an agent is deployed into an environment τ with a disturbance policy π_ϕ , from the perspective of the agent, the disturbance π_ϕ is an integral part of the environment τ . Thus, τ and ϕ jointly create an environment. We refer to such a new environment created by incorporating an adversary ϕ into an environment τ as a *variation environment*, denoted by (τ, ϕ) . Specifically,

the MDP of this environment can be defined as $(\mathcal{S}, \mathcal{A}, P_{\tau, \phi}, R_{\tau, \phi}, \gamma, \rho_{\tau})$, where $\mathcal{S}, \mathcal{A}, \gamma, \rho_{\tau}$ are defined as in the MDP formulation in Section 2, and $P_{\tau, \phi}(s, a) = \mathbb{E}_{a' \sim \pi_{\phi}(s)}[P_{\tau}^G(s, a, a')]$ and $R_{\tau, \phi}(s, a) = \mathbb{E}_{a' \sim \pi_{\phi}(s)}[R_{\tau}^G(s, a, a')]$ with P_{τ}^G and R_{τ}^G defined as in the Markov Game formulation in Section 3.1.

To pretrain TMs to generalize across \mathcal{M}_v , we sample a set of variation environments belonging to \mathcal{M}_v for supervised pretraining. To this end, given a set of m pretraining tasks $\{\tau^i\}_{i=1}^m \subset \mathcal{M}$, we create K variations of every pretraining task τ^i by randomly sampling K adversaries $\{\phi^{i,k}\}_{k=1}^K \subset \Phi(\tau^i)$ and including the sampled adversaries into τ^i . Thus, we create a set of variation environments $\{(\tau^i, \phi^{i,k})\}_{i \in [m], k \in [K]}$. For every variation environment, we sample an in-context dataset D , a query state s_{query} , and an optimal action label following the optimal policy $\pi_{\tau, \phi}^*(s_{\text{query}})$, defined as

$$\pi_{\tau, \phi}^* \in \arg \max_{\pi} \mathcal{R}_{\tau}(\pi, \phi). \quad (5)$$

We note that the objective function in Equation 5 does not involve max-min optimization. Moreover, learning optimal policies for K variation environments can be easily parallelized. This can significantly improve the efficiency the pretraining. See Algorithm 1 for its pseudocode.

4.2 IN-CONTEXT ADVERSARIAL ADAPTATION

The proposed ICAG framework has one notable limitation: obtaining optimal action labels for a variation environment (τ, ϕ) is computationally intensive, as it requires training a policy *from scratch* to approximate the optimal policy $\pi_{\tau, \phi}^*$ defined in Equation 5. To address this challenge, we propose leveraging a pretrained ICRL model to efficiently generate high-quality, though not necessarily optimal, action labels that are *sufficient to improve robustness* through iterative supervised updates.

We introduce *In-Context Adversarial Adaptation* (ICAA), a scalable method that uses an existing ICRL model to efficiently adapt to each variation environment $(\tau^i, \phi^{i,k})$ and generate query state–action label pairs for pretraining. Given an initial dataset \mathcal{D}^0 , we first train an initial model T_{θ}^0 with supervised pretraining on \mathcal{D}^0 and then deploy it in each variation environment. Here, \mathcal{D}^0 can be the pretraining dataset containing only trajectories from the original, unaugmented environments. Starting with an empty context buffer, the model performs online deployment by interacting with the environment and incrementally updating its context using its own rollouts, yielding a trajectory sequence $\xi_0, \xi_1, \dots, \xi_N$ (see Section 2 for details).

With these trajectories $\{\xi_n\}_{n=1}^N$, we construct a new extra pretraining dataset for each variation environment to update the current ICRL model T_{θ}^0 . Recall that an instance of the pretraining dataset $\{D, s_{\text{query}}, a^*\}$ has three elements: a context dataset D of trajectories, a query state s_{query} , and its corresponding action label a^* . To this end, we first collect the initial $\underline{N} < N$ trajectories $\{\xi_n\}_{n=0}^{\underline{N}}$ to construct the context datasets for pretraining. Next, we independently sample state–action pairs $\{s_{\text{query}}^n, a_n\}_{n=0}^{\underline{N}}$ from the remaining trajectories $\{\xi_n\}_{n=\underline{N}}^N$. The sampled pairs are used as the query state–action label pairs for pretraining. In particular, as the performance of ICRL models increases with more trajectories during deployment, we sample from later trajectories ξ_n where $n > \underline{N}$ to focus on trajectories with higher cumulative returns and obtain action labels with higher quality.

This yields a set of pretraining datasets $\{D_n = \xi_n, s_{\text{query}}^n, a_n\}_{n=0}^{\underline{N}}$ for each variation environment. Aggregating such datasets across all variations environments, we incorporate them into \mathcal{D}^0 to construct an augmented training set \mathcal{D}^1 . We then update the ICRL model by supervised pretraining on \mathcal{D}^1 to obtain T_{θ}^1 , which improves robustness and generalization *without requiring task-specific policies to be trained from scratch*. This process is repeated for J rounds to iteratively refine the model T_{θ}^J on \mathcal{D}^J . See Algorithm 2 for a detailed procedure and the ablation study on the number of model refinement in Dark Room environment in Appendix E.

Theoretical Analysis. We provide theoretical guarantees for both ICAG and ICAA to gain further insights into their efficacy. Due to space constraints, we present informal versions of our results to provide insights and defer our full results to Appendix F. We show two main results. Our most important result reveals that *well-pretrained ICAG models perform implicit Posterior Sampling over both the deployment task and the potential adversary during deployment*. In particular, Posterior Sampling (PS) is widely recognized as one of the most sample-efficient algorithms for a broad class of sequential decision-making problems (Osband & Van Roy, 2017). By implicitly performing PS

during deployment over *both* the deployment environment and potential adversarial perturbations, transformer models pretrained by ICAG are able to act optimally in the presence of an adversary capable of modifying the environment.

5 EXPERIMENTS

We evaluate our proposed methods on three benchmarks and compare them primarily against in-context learning baselines, including the Decision Pretrained Transformer (DPT) (Lee et al., 2024), Mixed Decision Pretrained Transformer (Mix-DPT) (Wang et al., 2024), Prompt-based Decision Transformer (Prompt-DT) (Xu et al., 2022), and Adversarially Robust Decision Transformer (ARDT) (Tang et al., 2024). To provide a broader assessment in online settings, we also include additional methods that are not in-context baselines but serve as complementary online evaluation references. Specifically, in Dark Room (Laskin et al., 2023) and Meta-World (Yu et al., 2020), we compare against Domain Randomization (DR) (Tobin et al., 2017) and M2TD3 (Tanabe et al., 2022), while in MuJoCo (Todorov et al., 2012) we additionally compare with SAC (Haarnoja et al., 2018), RARL (Pinto et al., 2017), and QARL (Reddi et al., 2024). We further demonstrate in Appendix E that ICAG generalizes more effectively under environmental changes. Full implementation and environment details are provided in Appendix C and Appendix D.

Computation and Implementation Efficiency. Our approach achieves efficient robustness by mitigating pre-training cost and avoiding expensive sequential max-min optimization. The core cost of data generation is managed by performing the entire MK dataset generation process using parallelized SAC runs, thereby avoiding the expensive sequential max-min procedure of conventional robust RL. All experiments were conducted on a single Nvidia RTX A5000 GPU with 24GB RAM. Once the data is collected, in DarkRoom, all transformer models converge within 150 epochs within an hour. In Meta-World, we run each experiment up to 500 episodes with early termination mechanism. In MuJoCo, all models can converge within 200 epochs with less than two hours.

5.1 DARK ROOM

Dark Room is a sparse-reward navigation task in a 10×10 2D discrete grid, where adversarial action is sampled from a Dirichlet distribution, turning the task into a probabilistic MDP. $K = 8$ is selected in MuJoCo based on ablation studies in Figure 7. For environment details and data generation, see Appendix D. Following the DPT evaluation protocol (Lee et al., 2024), we assess generalization to unseen tasks, by training on 80 goals and evaluating on 20 unseen goals. As shown in Figure 3(a), we report offline performance using both random and expert trajectories. All methods outperform the random trajectories (red), yet DT-based methods like prompt-DT and ARDT struggle with both random and expert test trajectories due to the randomness of the training set. In contrast, ICAG and ICAA outperform all baselines, including DPT, on both testing datasets, highlighting strong adaptation to expert-like behavior without direct exposure to expert trajectories. In Figure 3(b) and Figure 3(c), we evaluate online adaptation over 40 episodes. Figure 3(c) introduces environmental disturbances with unseen priors, modeled using a Dirichlet distribution with takeover probability $p = 0.2$. ICAG achieves the highest return, while ICAA and DPT gradually converge toward ICAG’s performance with increased adaptation. Under perturbations, ICAG’s advantage becomes more pronounced in Figure 3(c). Finally, Figure 4(a) evaluates robustness across varying disturbance probabilities. The x-axis denotes the probability of adversarial action override, while the y-axis shows the average return after 40 episodes. ICAG consistently outperforms all other methods *under disturbance*, maintaining robustness even under high disturbance level (e.g., $p = 0.6$).

5.2 META-WORLD

In the Meta-World benchmark, the agent is tasked with controlling a robotic hand to reach target positions in 3D space. The Meta-World adversary is a fixed-weight neural network that applies bounded magnitude perturbations ($|\phi^{i,k}| \leq 1$) directly to the observation space. This targets sensory inputs (pose, position) to simulate sensor uncertainty. Detailed descriptions of the environment and data generation process can be found in Appendix D. Meta-World includes 20 tasks in total. To assess the generalization capability of our approach to unseen reinforcement learning tasks, we train on 15 tasks and evaluate on the remaining 5. Offline performance is evaluated using both random and

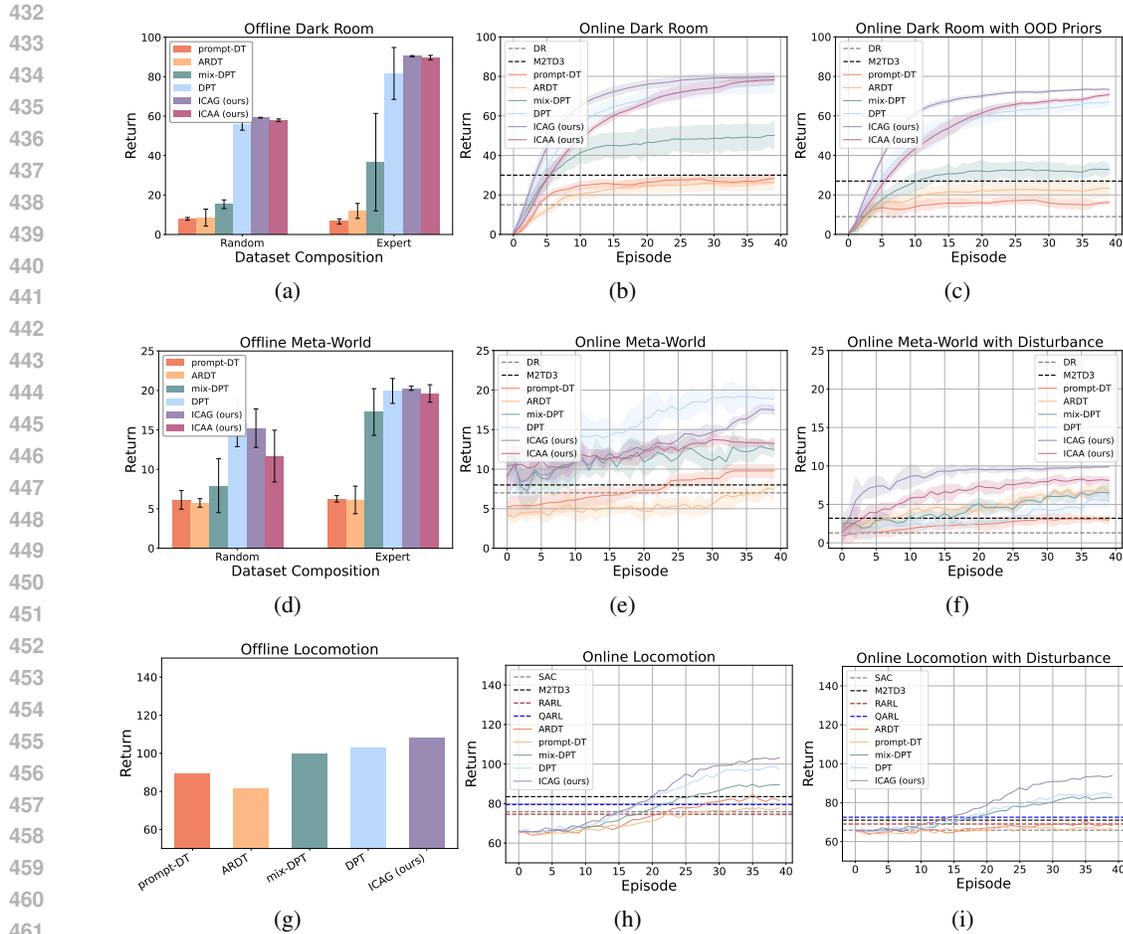


Figure 3: Performance on held-out Dark Room goals (top), held-out Meta-World goals (middle), and MuJoCo locomotion problems (bottom) with average return over 10 random seeds. The error bar and the shaded area represent the standard error. (1st column) Offline evaluation given random and expert datasets, or trajectories rollout from trained SAC (locomotion). (2nd and 3rd columns) Online evaluation without/with disturbances.

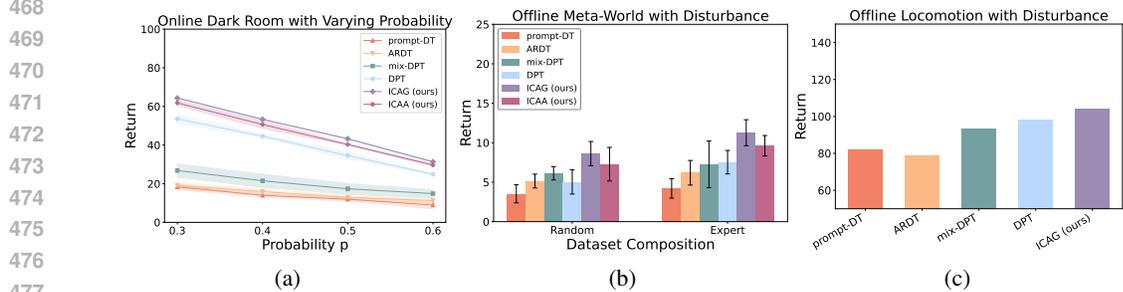


Figure 4: Additional experiments with average return over 10 random seeds. The error bar and the shaded area represent the standard error. (a) Online evaluation under disturbances with increasing probabilities. (b, c) Offline evaluation with disturbance given random and expert datasets, or trajectories rollout from trained SAC (locomotion).

expert trajectories, as shown in Figure 3(d) and Figure 4(b), while online performance is reported in Figure 3(e) and Figure 3(f). Across both offline and online settings, we observe that all methods exhibit performance degradation under state disturbances. Notably, while DPT suffers a significant

486 drop in performance, ICAG and ICAA maintain higher robustness, consistently outperforming other
 487 methods under perturbation.
 488

489 5.3 MUJoCo CONTROL

490 We evaluate our methods on a diverse set of continuous control tasks from the DeepMind Control
 491 Suite (Tunyasuvunakool et al., 2020), spanning 6 environments and 11 tasks (details in Appendix D).
 492 Pretraining datasets are constructed from historical trajectories generated by SAC agents. To ensure
 493 robustness during supervised pretraining in ICAG, we introduce K fixed adversaries $\{\phi^{i,k}\}_{k=1}^K$ for
 494 each task τ^i , where each adversary is modeled by a distinct neural network. **The MuJoCo adversary is
 495 based on active external forces applied within a zero-sum Markov Game framework. The adversary’s
 496 policy is parameterized to apply forces that destabilize the agent . The specific adversary action space
 497 and maximum force bounds are determined by domain knowledge and vary per environment.** SAC is
 498 trained to convergence under adversarial disturbances $\phi^{i,k}$, and the resulting trajectories are used to
 499 build variation environments $\{(\tau^i, \phi^{i,k})\}_{i \in [m], k \in [K]}$ for m pretraining tasks. **$K = 10$ is selected in
 500 MuJoCo based on ablation studies in Figure 8.** Additional dataset details are provided in Appendix D.
 501 We assess robustness from two complementary perspectives: (a) adversarial disturbances affecting the
 502 agent and (b) environmental variations, such as changes in mass and friction. Figure 3(g), 3(h), 3(i),
 503 4(c) summarizes performance on MuJoCo locomotion tasks, where our ICAG method consistently
 504 outperforms other baselines. Due to space constraints, the complete experimental results are provided
 505 in Appendix E. In brief, under adversarial disturbances ICAG achieves strong performance across
 506 both Cartpole and locomotion tasks, particularly in high-dimensional control problems such as
 507 Quadruped (see Figure 11 in Appendix E). Under environmental variations, ICAG likewise surpasses
 508 other baselines in out-of-distribution (OOD) settings, with the performance gap further widening
 509 compared to in-distribution results.
 510

511 6 RELATED WORKS

512 Owing to space limitations, we only review the most relevant work here and defer the comprehensive
 513 literature review to Appendix B. We position our work at the intersection of offline RL, transformer-
 514 based decision-making, and robust RL. Offline RL typically focuses on learning policies from fixed
 515 datasets for the same tasks, using value pessimism or policy regularization to address distributional
 516 shift. In contrast, ICRL aims to generalize to unseen tasks without parameter updates by leveraging
 517 transformers pretrained on diverse trajectories. While recent transformer-based methods (Chen et al.,
 518 2021; Xu et al., 2022; Lee et al., 2024; Laskin et al., 2023) demonstrate promising generalization,
 519 they often remain vulnerable to perturbations and out-of-distribution shifts. Robust and adversarial
 520 RL (Pinto et al., 2017; Tanabe et al., 2022) enhance resilience by optimizing worst-case performance
 521 or introducing disturbances during training, but frequently incur high computational costs or degrade
 522 in-distribution performance. Our proposed methods, ICAG and ICAA, integrate the strengths of both
 523 paradigms to improve robustness and adaptability in ICRL.
 524
 525
 526
 527

528 7 LIMITATIONS AND DISCUSSION

529 In this work, we study a practical link between generalization and robustness. Deploying pretrained
 530 ICRL models in robust-RL settings delivers *efficient robustness*: strong performance without the
 531 nominal-performance sacrifices of robustness-oriented methods or the computational burden of
 532 max–min optimization. However, this benefit comes with a structural limitation: ICRL requires a
 533 small context of trajectories at deployment (from offline logs or brief online rollouts). While such
 534 data are often easy to obtain, it may be impractical in safety-critical domains with severe data scarcity,
 535 where classical robust RL remains preferable. To strengthen both robustness and generalization in the
 536 ICRL regime, we propose two lightweight frameworks, ICAG and ICAA. ICAG emphasizes high
 537 performance under adversarial or perturbed conditions, whereas ICAA targets fast, data-efficient
 538 robustness improvements from minimal context. Practitioners can select between them based on
 539 compute and data budgets of their applications.

540 ETHICS STATEMENT
541

542 This study complies with the ICLR Code of Ethics. All datasets employed are publicly available and
543 open-source under licenses that permit research use. No private or personally identifiable information
544 was accessed, and no new data were collected from human subjects. The research does not pose
545 privacy, security, or fairness concerns. The authors declare no conflicts of interest and no external
546 sponsorship.

547
548 REPRODUCIBILITY STATEMENT
549

550 All datasets used in our experiments are publicly available and described in Section 5 and Ap-
551 pendix D.1. Implementation details of baselines, model architecture and hyper-parameter settings are
552 provided in Appendix C. We also attach our codebase as Supplementary Material. Complete proofs
553 of theoretical results and assumptions underlying our analysis are included in Appendix G. These
554 resources together allow independent researchers to verify and reproduce our findings.

555
556 REFERENCES
557

- 558 Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal,
559 Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are
560 few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- 561 Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent
562 reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications
563 and Reviews)*, 38(2):156–172, 2008.
- 564 Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel,
565 Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence
566 modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.
- 567 Juncheng Dong, Weibin Mo, Zhengling Qi, Cong Shi, Ethan X Fang, and Vahid Tarokh. Pasta:
568 pessimistic assortment optimization. In *International Conference on Machine Learning*, pp.
569 8276–8295. PMLR, 2023.
- 570 Juncheng Dong, Moyang Guo, Ethan X Fang, Zhuoran Yang, and Vahid Tarokh. In-context reinforce-
571 ment learning from suboptimal historical data. In *International conference on machine learning*.
572 PMLR, 2025a.
- 573 Juncheng Dong, Hao-Lun Hsu, Qitong Gao, Vahid Tarokh, and Miroslav Pajic. Variational adversarial
574 training towards policies with improved robustness. In *Artificial intelligence and statistics*. PMLR,
575 2025b.
- 576 Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning.
577 *Advances in neural information processing systems*, 34:20132–20145, 2021.
- 578 Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy
579 maximum entropy deep reinforcement learning with a stochastic actor. In *International conference
580 on machine learning*, pp. 1861–1870. PMLR, 2018.
- 581 Garud N. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):
582 257–280, 2005.
- 583 Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In
584 *International Conference on Machine Learning*, pp. 5084–5096. PMLR, 2021.
- 585 Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey.
586 *Journal of artificial intelligence research*, 4:237–285, 1996.
- 587 Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-
588 based offline reinforcement learning. *Advances in neural information processing systems*, 33:
589 21810–21823, 2020.

- 594 Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline
595 reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.
- 596
- 597 Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald,
598 DJ Strouse, Steven Hansen, Angelos Filos, Ethan Brooks, et al. In-context reinforcement learning
599 with algorithm distillation. *International Conference on Learning Representations*, 2023.
- 600
- 601 Jonathan Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma
602 Brunskill. Supervised pretraining can learn in-context reinforcement learning. *Advances in Neural
603 Information Processing Systems*, 36, 2024.
- 604
- 605 Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial,
606 review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.
- 607
- 608 Wenzhe Li, Hao Luo, Zichuan Lin, Chongjie Zhang, Zongqing Lu, and Deheng Ye. A survey
609 on transformers in reinforcement learning. *Transactions on Machine Learning Research*, 2023.
610 ISSN 2835-8856. URL <https://openreview.net/forum?id=r30yuDPvf2>. Survey
611 Certification.
- 612
- 613 Licong Lin, Yu Bai, and Song Mei. Transformers as decision makers: Provable in-context reinforce-
614 ment learning via supervised pretraining. In *The Twelfth International Conference on Learning
615 Representations*, 2024. URL <https://openreview.net/forum?id=yN4Wv17ss3>.
- 616
- 617 Zuxin Liu, Zijian Guo, Yihang Yao, Zhepeng Cen, Wenhao Yu, Tingnan Zhang, and Ding Zhao. Con-
618 strained decision transformer for offline safe reinforcement learning. In *International Conference
619 on Machine Learning*, pp. 21611–21630. PMLR, 2023.
- 620
- 621 Tatsuya Matsushima, Hiroki Furuta, Yutaka Matsuo, Ofir Nachum, and Shixiang Gu. Deployment-
622 efficient reinforcement learning via model-based offline optimization. *arXiv preprint
623 arXiv:2006.03647*, 2020.
- 624
- 625 Janosch Moos, Kay Hansel, Hany Abdulsamad, Svenja Stark, Debora Clever, and Jan Peters. Robust
626 reinforcement learning: A review of foundations and recent advances. *Machine Learning and
627 Knowledge Extraction*, 4(1):276–315, 2022.
- 628
- 629 Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain
630 transition matrices. *Operations Research*, 53(5):780–798, 2005.
- 631
- 632 OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni
633 Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor
634 Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian,
635 Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny
636 Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks,
637 Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea
638 Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen,
639 Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung,
640 Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch,
641 Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty
642 Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte,
643 Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel
644 Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua
645 Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike
646 Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon
647 Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne
Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo
Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar,
Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik
Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich,
Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy
Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie
Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini,
Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne,

- 648 Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David
649 Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie
650 Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély,
651 Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo
652 Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano,
653 Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng,
654 Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto,
655 Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power,
656 Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis
657 Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted
658 Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel
659 Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon
660 Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky,
661 Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang,
662 Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston
663 Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya,
664 Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason
665 Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff,
666 Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu,
667 Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba,
668 Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang,
669 William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.
- 669 Ian Osband and Benjamin Van Roy. Why is posterior sampling better than optimism for reinforcement
670 learning? In *International conference on machine learning*, pp. 2701–2710. PMLR, 2017.
- 671 Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial rein-
672 forcement learning. In *International conference on machine learning*, pp. 2817–2826. PMLR,
673 2017.
- 674 Rafael Figueiredo Prudencio, Marcos ROA Maximo, and Esther Luna Colombini. A survey on offline
675 reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural*
676 *Networks and Learning Systems*, 2023.
- 677 Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language
678 models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- 679 Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. Epop: Learning
680 robust neural network policies using model ensembles. *International Conference on Learning*
681 *Representations*, 2017.
- 682 Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging offline rein-
683 forcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information*
684 *Processing Systems*, 34:11702–11716, 2021.
- 685 Aryaman Reddi, Maximilian Tölle, Jan Peters, Georgia Chalvatzaki, and Carlo D’Eramo. Robust
686 adversarial reinforcement learning via bounded rationality curricula. *International Conference on*
687 *Learning Representations*, 2024.
- 688 Franco Scarselli and Ah Chung Tsoi. Universal approximation using feedforward neural networks: A
689 survey of some existing methods, and some new results. *Neural networks*, 11(1):15–37, 1998.
- 690 Takumi Tanabe, Rei Sato, Kazuto Fukuchi, Jun Sakuma, and Youhei Akimoto. Max-min off-policy
691 actor-critic method focusing on worst-case robustness to model misspecification. In Alice H. Oh,
692 Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information*
693 *Processing Systems*, 2022.
- 694 Xiaohang Tang, Afonso Marques, Parameswaran Kamalaruban, and Ilija Bogunovic. Adversarially
695 robust decision transformer. *Advances in neural information processing systems*, 2024.
- 696 Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applica-
697 tions in continuous control. In *International Conference on Machine Learning*, pp. 6215–6224.
698 PMLR, 2019.

- 702 Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Mujoco:
703 A physics engine for model-based control. In *2017 IEEE/RSJ international conference on intelligent*
704 *robots and systems (IROS)*, pp. 23–30. IEEE, 2017.
- 705 Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control.
706 In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033.
707 IEEE, 2012.
- 708 Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei,
709 Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas
710 Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes,
711 Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S.
712 Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa,
713 Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril,
714 Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar
715 Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan
716 Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang,
717 Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen
718 Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic,
719 Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models.
720 *ArXiv*, abs/2307.09288, 2023. URL <https://api.semanticscholar.org/CorpusID:259950998>.
- 721
- 722 Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom
723 Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm control: software and tasks for
724 continuous control. *Software Impacts*, 6, 2020.
- 725
- 726 Hanzhao Wang, Yu Pan, Fupeng Sun, Shang Liu, Kalyan Talluri, Guanting Chen, and Xiaocheng Li.
727 Understanding the training and generalization of pretrained transformer for sequential decision
728 making. *arXiv preprint arXiv:2405.14219*, 2024.
- 729 Tianyu Wu, Shizhu He, Jingping Liu, Siqi Sun, Kang Liu, Qing-Long Han, and Yang Tang. A brief
730 overview of chatgpt: The history, status quo and potential future development. *IEEE/CAA Journal*
731 *of Automatica Sinica*, 10(5):1122–1136, 2023.
- 732
- 733 Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning,
734 2019.
- 735 Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context
736 learning as implicit bayesian inference. In *International Conference on Learning Representations*,
737 2022. URL <https://openreview.net/forum?id=RdJVFCHjUMI>.
- 738 Mengdi Xu, Yikang Shen, Shun Zhang, Yuchen Lu, Ding Zhao, Joshua Tenenbaum, and Chuang Gan.
739 Prompting decision transformer for few-shot policy generalization. In *international conference on*
740 *machine learning*, pp. 24631–24645. PMLR, 2022.
- 741
- 742 Taku Yamagata, Ahmed Khalil, and Raul Santos-Rodriguez. Q-learning decision transformer:
743 Leveraging dynamic programming for conditional sequence modelling in offline rl. In *International*
744 *Conference on Machine Learning*, pp. 38989–39007. PMLR, 2023.
- 745
- 746 Ming Yin and Yu-Xiang Wang. Towards instance-optimal offline reinforcement learning with
747 pessimism. *Advances in neural information processing systems*, 34:4065–4078, 2021.
- 748
- 749 Chengyang Ying, Xinning Zhou, Hang Su, Dong Yan, Ning Chen, and Jun Zhu. Towards safe
750 reinforcement learning via constraining conditional value at risk. *International Joint Conference*
751 *on Artificial Intelligence*, 2022.
- 752
- 753 Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey
754 Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning.
755 In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.
- 756
- 757 William Yuan, Jiaying Chen, Shaofei Chen, Lina Lu, Zhenzhen Hu, Peng Li, Dawei Feng, Furong Liu,
758 and Jing Chen. Transformer in reinforcement learning for decision-making: A survey. *Authorea*
759 *Preprints*, 2023.

756 Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui
757 Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations.
758 *Advances in Neural Information Processing Systems*, 33:21024–21037, 2020.
759

760 Qinqing Zheng, Amy Zhang, and Aditya Grover. Online decision transformer. In *international*
761 *conference on machine learning*, pp. 27042–27059. PMLR, 2022.

762 Ilya Zisman, Vladislav Kurenkov, Alexander Nikulin, Viacheslav Sinii, and Sergey Kolesnikov.
763 Emergence of in-context reinforcement learning from noise distillation. In *International conference*
764 *on machine learning*. PMLR, 2024.
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809

810 A USE OF LARGE LANGUAGE MODELS (LLMs)

811
812 Large Language Models (LLMs) were used solely as general-purpose assistive tools to improve the
813 clarity and readability of the manuscript. Specifically, we used an LLM to help rephrase and polish
814 text that we had already drafted.

815 816 B RELATED WORK

817
818
819 **Offline RL.** Our work is situated within the broader field of offline RL (Levine et al., 2020; Mat-
820 sushima et al., 2020; Prudencio et al., 2023). Offline RL methods often employ approaches such
821 as value pessimism or policy regularization to address the distributional shift between behavioral
822 and optimal policies (Wu et al., 2019; Kidambi et al., 2020; Kumar et al., 2020; Rashidinejad et al.,
823 2021; Yin & Wang, 2021; Jin et al., 2021; Dong et al., 2023; Fujimoto & Gu, 2021). While offline RL
824 aims to solve the *same* tasks from which the offline datasets are collected, ICRL aims to efficiently
825 generalize and adapt to *unseen* tasks.

826 **Transformers for Decision Making.** Autoregressive models (Radford et al., 2019; Brown et al.,
827 2020; Wu et al., 2023; Touvron et al., 2023; OpenAI et al., 2024) have achieved remarkable successes
828 across various domains. Their application to sequential decision-making tasks, such as bandit and
829 MDP problems, has shown transformers outperforming traditional methods (Li et al., 2023; Yuan
830 et al., 2023). Decision Transformer (DT) (Chen et al., 2021; Zheng et al., 2022; Liu et al., 2023;
831 Yamagata et al., 2023) formulates offline RL as return-conditioned supervised learning and scales
832 well across multi-task settings. Methods like prompt-DT (Xu et al., 2022), Algorithm Distillation
833 (AD) (Laskin et al., 2023), and DPT (Lee et al., 2024) aim to enhance in-context generalization to new
834 goals and tasks. Recent efforts (Dong et al., 2025a; Zisman et al., 2024) further relax the assumptions
835 on pretraining datasets by leveraging suboptimal trajectories or policies. However, challenges remain
836 in addressing out-of-distribution (OOD) contexts with different environment dynamics and robustness
837 to disturbances. To tackle these issues, we propose a transformer-based approach that leverages
838 in-context learning to address the robustness problem and further utilize online adaptation to learn
839 high-quality action without training separate policies for each adversarial task from scratch.

840 **Robust RL and Adversarial Training.** Robust reinforcement learning (RL) focuses on generalizing
841 to out-of-distribution (OOD) environments by optimizing worst-case performance across various
842 transition models (Nilim & Ghaoui, 2005; Iyengar, 2005). Deep RL methods commonly achieve
843 robustness during training through parametric uncertainty, which considers a range of simulation
844 parameters to optimize for worst-case performance (Tanabe et al., 2022; Rajeswaran et al., 2017),
845 or through adversarial training, which introduces perturbations to the environment (e.g., actions,
846 observations, or transitions) to simulate potential deployment-time disturbances (Pinto et al., 2017;
847 Tessler et al., 2019; Zhang et al., 2020; Reddi et al., 2024). These methods enhance generalization to
848 dynamics unseen during training but often sacrifice in-distribution (ID) performance.

849 C IMPLEMENTATION DETAILS

850 C.1 ALGORITHM.

851
852 **Soft Actor-Critic (SAC) (Haarnoja et al., 2018).** Soft Actor-Critic (SAC) is an off-policy rein-
853 forcement learning algorithm designed to balance exploration and exploitation by optimizing the
854 trade-off between expected rewards and action entropy. SAC aims to learn a policy that not only
855 maximizes long-term rewards but also encourages exploration by maximizing the entropy of the
856 policy’s action distribution. The algorithm utilizes an actor network to select actions, along with two
857 critic networks that estimate the Q -values of state-action pairs. The learning objective is to maximize
858 a soft Bellman equation: $J(\pi) = \sum_h \mathbb{E}_{(s_h, a_h) \sim D} [Q(s_h, a_h) - \alpha \log \pi(a_h | s_h)]$, where $Q(s_h, a_h)$
859 represents the value of the state-action pair as estimated by the critics, α is a temperature parameter
860 controlling the exploration-exploitation balance, and $\pi(a_h | s_h)$ is the probability of selecting action
861 a_h in state s_h . SAC is trained by sampling mini-batches from a replay buffer to update both the policy
862 and the Q -value estimates. We use the implementation without adversarial training from Reddi et al.
863 (2024).

Decision-Pretrained Transformer (DPT) (Lee et al., 2024). The Decision-Pretrained Transformer (DPT) is designed for in-context learning in reinforcement learning (RL) tasks by leveraging supervised pretraining. The key idea behind DPT is to train a transformer model to predict optimal actions for a given query state, using an in-context dataset that includes interactions from a variety of tasks. These interactions are represented as state-action-reward tuples, which provide the necessary context for decision-making. During pretraining, DPT samples a distribution of tasks. For each task τ_i , an in-context dataset D^i , is constructed, consisting of sequences of state-action-reward transitions that reflect prior experience with that task. A query state s_{query}^i is then sampled from the MDP’s state space, and the model is trained to predict the optimal action based on both the query state and the task-specific context D^i . The training objective is to minimize the expected loss over the sampled task distribution, where the model learns to predict a distribution of actions given the query state and context. DPT shares the same set of finite environments as ICAG and ICAA, but its pretraining dataset does not consider robustness.

Mixed Decision-Pretrained Transformer (mix-DPT) (Wang et al., 2024). The mix-DPT framework extends DPT by splitting the learning process into two phases: the early training phase and the mixed training phase, addressing the out-of-distribution (OOD) issue between training and testing. During the early training phase, data are generated using a pre-specified decision function f , such as a random policy for the Dark Room task or SAC-trained policies for Meta-World and MuJoCo control. In the mixed training phase, data are generated using both the function f and the current DPT model, with the proportion controlled by a hyper-parameter κ . In the experiments conducted by Wang et al. (2024), historical trajectories include the optimal actions for each time step. To ensure a fair comparison, we follow their setup but use only a query state and the corresponding optimal action, as done in DPT and ICAG. For our experiments, we follow the parameter choice $\kappa = 1/3$ from Wang et al. (2024), with the total number of training iterations set to 40% of the overall training iterations.

Prompt-based Decision Transformer (prompt-DT) (Xu et al., 2022). Prompt-DT builds upon Decision Transformer (Chen et al., 2021) and organizes its data to enable few-shot policy generalization through the use of trajectory prompts. For each task T_i , a prompt τ_i^* of length K^* is constructed from a set of few-shot demonstrations P_i , which consist of state-action-reward-to-go tuples (s^*, a^*, \hat{G}^*) . This prompt captures the task-specific context required for policy adaptation. To further enrich the context, the most recent trajectory history τ_i , sampled from an offline dataset D_i , is appended to the task-specific prompt. This forms the complete input sequence τ_{input} input. Specifically, the input sequence is represented as: $\tau_{\text{input}} = (\tau_i^*, \tau_i)$. This sequence consists of $3(K^* + K)$ tokens, following the state-action-reward format. The full sequence τ_{input} is then processed by a Transformer model, which autoregressively predicts the next actions corresponding to each state token. We follow the original prompt-DT setup and set $k = 20$. Prompt-DT uses the same pretraining dataset as DPT, but lacks query state-action pairs, highlighting the architecture’s effectiveness in DPT-based methods (e.g., DPT, ICAG, ICAA).

Adversarially Robust Decision Transformer (ARDT) (Tang et al., 2024). ARDT enhances the Decision Transformer by associating worst-case returns-to-go via minimax expectile regression with trajectories to improve robustness against adversarial perturbations. Specifically, the estimated Q values from expectile regression replaces the returns-to-go in vanilla DT during training.

Robust Adversarial Reinforcement Learning (RARL) (Pinto et al., 2017). RARL trains a protagonist to compete against destabilizing forces introduced by an adversary in a zero-sum Markov game, where the optimal strategy (i.e., the rational strategy) corresponds to a Nash equilibrium. In this setup, the protagonist selects actions to maximize performance, while the adversary is trained to take actions that minimize the same performance metric. By training under these destabilizing perturbations, the protagonist learns to develop robust skills that help it counter distribution shifts and adversarial attacks when deployed in real-world scenarios. We implement RARL using the framework from Reddi et al. (2024), where both the protagonist and adversary are represented as agents with SAC policies.

Quantal Adversarial Reinforcement Learning (QARL) (Reddi et al., 2024). QARL formulates a robust adversarial reinforcement learning objective with entropy regularization, designed to model

918 Markov games under bounded rationality. It introduces two temperature coefficients for the Shannon
 919 entropy of both the protagonist’s and adversary’s action distributions, allowing the optimization
 920 problem between the two players to be framed as a Quantal Response Equilibrium (QRE). QRE is
 921 a generalization of the Nash equilibrium, extending it to scenarios where agents may not act with
 922 complete rationality. We implement QARL using the framework from Reddi et al. (2024), where
 923 both the protagonist and adversary are represented by SAC-based agents.

924 C.2 MODEL ARCHITECTURE AND HYPER-PARAMETERS

925 **Transformer.** Our models, DPT, MDPT, and prompt-DT are all based on a causal GPT-2 archi-
 926 tecture (Radford et al., 2019). It consists of 4 attention layers, each with a single attention head. In
 927 DarkRoom, the embedding size is 32, while Meta-World and MuJoCo environments’ embedding
 928 size are 256. Prompt-DT and ARDT are built on Decision Transformer, separating the individual
 929 (s, a, s', r) into their own embeddings to be made into one long sequence. The remaining transformer-
 930 based baselines and our models view the transition tuples in the dataset as their own singletons, to be
 931 related with other singletons in the dataset through the attention mechanism. We use the AdamW
 932 optimizer with a weight decay of $1e - 4$, a learning rate of $1e - 3$, and a batch size of 128.

933 **Multilayer perceptron (MLP).** For all non-transformer agents, e.g., SAC, RARL, QARL, we
 934 directly list their shared hyper-parameters and architecture in Table 1.

935 Table 1: Non-transformer agent hyper-parameters (e.g, SAC, RARL, QARL)

Hyper-parameters	Values
No of hidden layers	3
No of hidden units per layer	256
activation function	ReLU
optimizer (actor and critic)	Adam
actor learning rate	$1 \cdot 10^{-4}$
critic learning rate	$3 \cdot 10^{-4}$
initial replay memory size	$3 \cdot 10^3$
max replay memory size	$1 \cdot 10^6$
warmup transitions	$5 \cdot 10^3$
batch size	256
target smoothing coefficient	$5 \cdot 10^{-3}$
initial temperature	$5 \cdot 10^3$
temperature learning rate	$3 \cdot 10^{-4}$

955 D ENVIRONMENT SETTINGS AND PRETRAINING DATASET

956 D.1 DARKROOM.

957 Dark Room is a sparse-reward navigation task set in a discrete 10×10 grid. At the beginning of each
 958 episode, the agent is randomly placed in one of the grid cells, while the goal location is hidden and
 959 fixed at a random cell. The agent receives an observation of its current (x, y) position and selects from
 960 five discrete actions: move left, right, up, down, or stay in place. The episode horizon is $H = 100$
 961 steps. The agent receives a reward of $r = 1$ only when it reaches the goal, and $r = 0$ otherwise. At
 962 test time, the agent always starts from the origin $(0, 0)$.

963 To construct the pretraining dataset, we generate 100,000 trajectories using a uniform-random policy,
 964 evenly distributed across 100 different goal locations. For each, we sample query states uniformly
 965 and compute optimal actions using a heuristic that first aligns the agent’s y -coordinate with the goal
 966 and then the x -coordinate. The dataset is partitioned into 80,000 training examples (corresponding
 967 to 80 goal locations) and 20,000 validation examples (corresponding to the remaining 20 goals).

968 To improve robustness, we augment the environment with adversarial perturbations. Specifically,
 969 for each of the m training tasks $\{\tau^i\}_{i=1}^m$, we construct a set of perturbed environments $\mathcal{M}_v =$
 970

972 $\{(\tau^i, \phi^{i,k})\}_{i \in [m], k \in [K]}$ by allowing an adversarial action to override the agent’s intended action with
 973 probability $p \sim \mathcal{U}(0.0, 0.2)$. The adversarial action is sampled from a Dirichlet distribution by priors
 974 α , with the most probable action selected. With probability $1 - p$, the agent executes its original
 975 action. This construction models each perturbed task as a probabilistic MDP (Tessler et al., 2019),
 976 capturing structured uncertainty in action dynamics.

977 The Dirichlet distribution used satisfies $\sum_{i=1}^I x_i = 1$ and $x_i \in [0, 1]$ for all i . The probability density
 978 function is given by:

$$981 f(x_1, \dots, x_I; \alpha_1, \dots, \alpha_I) = \frac{1}{\beta(\alpha)} \prod_{i=1}^I x_i^{\alpha_i-1}.$$

982 , where $\beta(\alpha)$ is the multivariate beta function, which can be expressed in terms of the gamma function

$$983 \beta(\alpha) = \frac{\prod_{i=1}^I \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^I \alpha_i)}, \quad \alpha = (\alpha_1, \dots, \alpha_I).$$

984 These perturbed environments are used to generate additional trajectories in the style of DPT,
 985 increasing the diversity and robustness of the pretraining dataset.

986 D.2 META-WORLD.

987 We focus on the ML1 pick-place benchmark, where the agent must grasp an object and place it at
 988 a designated target location. Each task is defined by a 39-dimensional state space that includes the
 989 gripper’s position and binary open/close state, the 3D pose of the object, and the coordinates of the
 990 target. The agent operates in a continuous action space, allowing it to adjust its end-effector position
 991 in three dimensions and control the gripper’s open/close state to facilitate object manipulation.

992 The environment provides shaped rewards to guide learning, including incentives for approaching
 993 the object, establishing a grasp, transporting the object, and releasing it at the target. The goal is
 994 to learn a policy that effectively sequences these skills to solve the overall manipulation task. For
 995 generalization evaluation, we train on 15 task configurations and test on 5 held-out tasks with novel
 996 object and target positions.

997 To construct the pretraining dataset, we collect historical trajectories from agents trained using Soft
 998 Actor-Critic (SAC). For each task, SAC is trained until convergence, and we sample from the resulting
 999 trajectories to form offline datasets. The built-in deterministic policy is treated as the optimal expert
 1000 policy.

1001 To simulate real-world deployment challenges, we introduce robustness through adversarial perturba-
 1002 tions to the observation space. Specifically, for each task τ^i , we construct K perturbed variations
 1003 $\{(\tau^i, \phi^{i,k})\}_{i \in [m], k \in [K]}$, where each $\phi^{i,k}$ is implemented as a fixed-weight neural network with
 1004 bounded magnitude ($|\phi^{i,k}| \leq 1$). These perturbations target only the perceptually estimated com-
 1005 ponents of the observation—namely, object pose, end-effector position, and goal location—while
 1006 avoiding internal robot states such as joint angles or velocities, which are typically less affected by
 1007 real-world noise. This design mimics sensor uncertainty (e.g., from depth cameras or visual drift)
 1008 without destabilizing control signals that are precise on physical hardware.

1009 D.3 MUJoCo.

1010 The MuJoCo environments (Todorov et al., 2012) used in our experiments are standard implementa-
 1011 tions from the DeepMind Control Suite (Tunyasuvunakool et al., 2020). Specifically, we consider
 1012 modified versions of these environments that incorporate adversarial settings (i.e., RARL, QARL, and
 1013 pretraining for ICAG). In each environment, adversarial actions and force magnitudes are carefully
 1014 selected to challenge the agent and promote robust behavior. The adversarial action spaces are
 1015 intentionally designed to differ from those of the protagonist agent to exploit domain knowledge. The
 1016 adversary forces are calibrated to be sufficiently large to foster agent robustness and generalization,
 1017 while still posing a significant challenge to the protagonist.

Table 2: MuJoCo Environment-specific parameters for adversarial training and (protagonist) agent’s standard observation/action space

Environment	Adversary max force	Adversary action space	Observation space	Action space
Cartpole	0.005	2D forces on pole (1)	5	1
Cheetah	1.0	2D force on feet & torso (6)	17	6
Hopper	1.0	2D force on feet & torso (4)	15	4
Quadruped	10	3D force on torso (1)	78	12
Reacher	0.1	2D force on arm (2)	6	2
Walker	1.0	2D force on feet (4)	24	6

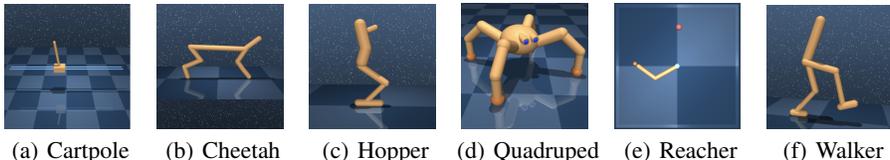


Figure 5: Illustrations of the MuJoCo environments.

We conduct experiments in 6 environments shown in Figure 5. Then environment-specific parameters, along with the corresponding observation and action spaces for the standard environments, are detailed in Table 2. For all environments, the discount factor is set to 0.9 and the horizon is set to 200.

Each environment is associated with one or more problems, defined as instances of the model with specific Markov Decision Process (MDP) structures. For example, in the CartPole environment, we define two problems: **swingup**, where the pole starts pointing downward, and **balance**, where the pole begins near the upright position. The goal in both problems is to manipulate the forces applied to a cart at the base in order to either swing up or balance an unactuated pole. In the Reacher environment, a two-link planar arm must reach a randomized target location, with a reward of 1 when the end effector reaches the target sphere. Two problems are defined here: in the **easy** problem, the target sphere is larger than in the **hard** problem.

In the Hopper environment, a planar one-legged hopper is initialized in a random configuration and is rewarded based on torso height and forward velocity. The remaining three environments: Walker, Cheetah, and Quadruped focus on maximizing forward velocity. In the Cheetah environment, the reward is linearly proportional to the forward velocity, capped at a maximum of $10m/s$. The Walker environment includes two tasks: **walk** and **run**, which differ in their velocity requirements and include components to encourage upright posture and minimal lateral movement. For standard in-context reinforcement learning (ICRL) methods, such as DPT (Lee et al., 2024), We use m pretraining tasks $\{\tau^i\}_{i=1}^m \subset \mathcal{M}$ with varying internal conditions. For each task τ^i , we construct a pretraining dataset using 6000 historical trajectories collected from SAC agents trained to convergence.

We generate K variations of each pretraining task τ^i , incorporating adversaries $\phi^{i,k}$ to form a set of variation environments $\{(\tau^i, \phi^{i,k})\}_{i \in [m], k \in [K]}$. In our experiments, $m = 16$ tasks (from a 4×4 grid) and $K = 10$ variations, with hyper-parameter tuning in Appendix E.3.

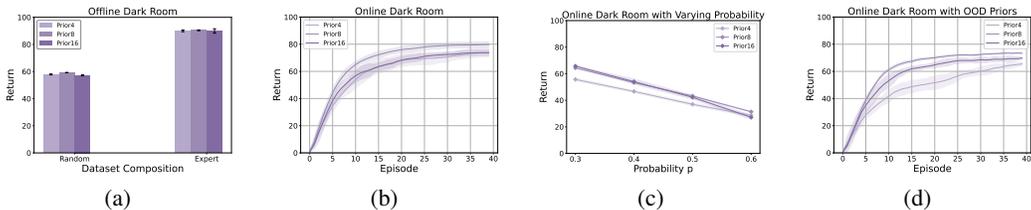
We follow the SAC training approach in Reddi et al. (2024), training $K = 10$ SAC policies for each τ^i under different variations, which are treated as distinct tasks. Each variation is initialized as a fixed-weight neural network ($|\phi^{i,k}| \leq 1$), unlike RARL or QARL, where adversarial policies are updated during training. These variations act as adversaries, with SAC policies trained under disturbance.

E ADDITIONAL EXPERIMENTAL RESULTS

E.1 ABLATION ON NUMBER OF PRIORS FOR ADVERSARIES IN DARK ROOM

We conduct ablation studies to assess the impact of varying the number of adversaries $\{\phi^{i,k}\}_{k=1}^K$ with different Dirichlet priors in the pretraining dataset to ICAG. The total number of pretraining

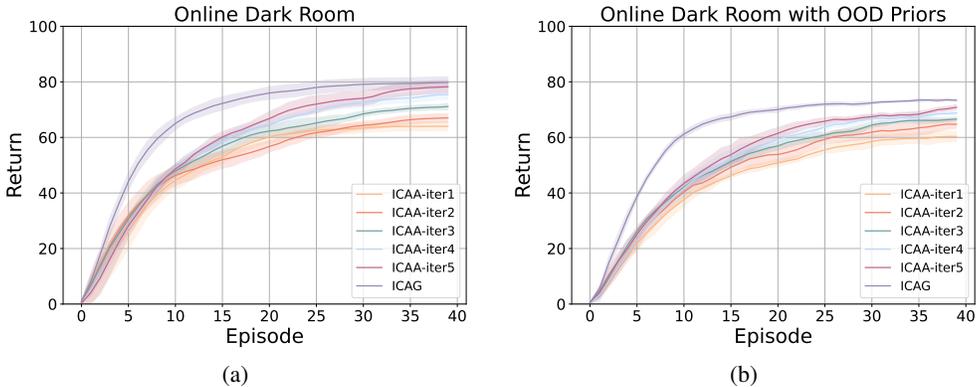
1080
1081
1082
1083
1084
1085
1086
1087



1088
1089
1090
1091
1092
1093
1094

Figure 6: Ablation study for ICAG with different number of training priors on held-out Dark Room goals from test tasks with average return over 10 random seeds. The error bar and the shaded area represent the standard error. (a) Offline evaluation given random and expert datasets. (b) Online evaluation without disturbances. (c) Online evaluation under disturbances with higher probability. (d) Online evaluation under disturbances with unseen priors. Note that prior $K \in \{4, 8, 16\}$ in each subfigure implies K adversaries involved in pretraining dataset.

1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107



1108
1109
1110
1111
1112
1113

Figure 7: Performance on held-out Dark Room goals with average return over 10 random seeds among our methods. Note that ICAA-iter1 denotes the online evaluation with T_{θ}^1 . The error bar and the shaded area represent the standard error. (a) Online evaluation without disturbances. (b) Online evaluation under disturbances with unseen priors.

1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126

trajectories remains consistent across different priors. Specifically, we explore $K \in \{4, 8, 16\}$ with varying α values (see Appendix D.1), and present the results in Figure 6.

In offline evaluation (shown in Figure 6(a)), both random and expert trajectories yield similar performance across different K values. For online evaluation, Figure 6(b) shows results without disturbances, while Figure 6(d) introduces disturbances with a probability of $p = 0.2$, which were absent during pretraining. When fewer adversaries are used (e.g., $K = 4$), the agent requires more episodes to adapt to environments with unseen priors, as indicated in Figure 6(d).

Finally, we evaluate all methods with a single adversary using a uniform distribution for disturbances, varying the disturbance probability in Figure 6(c). Based on these results, we select $K = 8$ for a better balance between online and offline performance, and use it for comparisons with other baselines in Figure 3.

1127
1128
1129

E.2 ABLATION ON THE NUMBER OF ICAA ROUNDS IN DARK ROOM

1130
1131
1132
1133

In Section 4.2 and Algorithm 2, ICAA employs an online approach to generate high-quality action labels and iteratively refine the model T_{θ}^J using the dataset \mathcal{D}^J , thereby reducing dependence on optimal action labels. We present the online evaluation results both with and without disturbances in Figure 7, and compare them against ICAG. Here, ICAA-iter1 represents the online evaluation using the model T_{θ}^1 . The results demonstrate that increasing the number of refinement iterations consistently

1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187

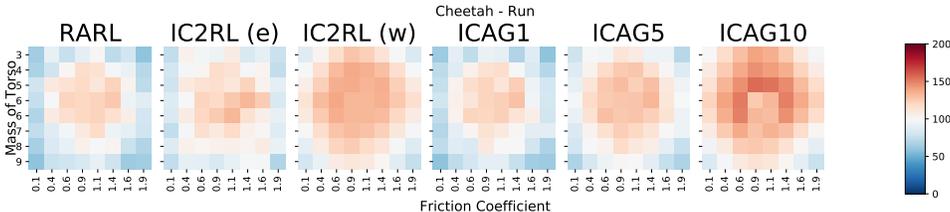


Figure 8: Generalization capability mainly among IC2RL and ICAG with varying K variations, where ICAG K , with $K \in \{1, 5, 10\}$.

improves performance under both without and with disturbances. However, the performance of ICAA remains bounded above by ICAG, as ICAG directly leverages optimal action labels.

E.3 VARIANT OF ICAG AND IC2RL

As discussed in Section 3.2, extending traditional robust adversarial RL approaches to in-context settings presents challenges. To explore this, we conduct experiments in the Cheetah environment, where for each task τ^i with 16 pairs of internal conditions, we learn a robust policy via RARL, as shown in Figure 2(b).

In Figure 8, we denote the well-converged variant of IC2RL as IC2RL(w), demonstrating improvement against RARL, but it is relatively impractical, solving max-min optimizations for all tasks. Furthermore, to directly test the sensitivity of IC2RL to the quality of its action labels, we introduce an additional variant, IC2RL(e). This model uses labels generated from early-stopped RARL policies (checkpoint saved at approximately half of the learning steps until convergence). The performance of IC2RL(e) is significantly degraded compared to IC2RL(w), confirming that IC2RL performance is fundamentally bounded by the quality and high computational cost of its robust label-generating policies.

Additionally, we investigate the effect of varying the number of adversaries K in $\{(\tau^i, \phi^{i,k})\}_{i \in [m], k \in [K]}$. We show that increasing K acts as a data augmentation technique, supporting our statement in Section 4.1 that the augmented task space \mathcal{M}_v covers the original task space \mathcal{M} ($\mathcal{M} \subset \mathcal{M}_v$), leading to better generalization. Notably, ICAG10 with $K = 10$, as shown in Figure 8, outperforms IC2RL.

E.4 MUJoCo PERFORMANCE AND ROBUSTNESS

Notably, we use SAC agents in an adversarial training framework in both RARL and QARL, making SAC a natural baseline. Our adversarial setup follows (Reddi et al., 2024), with environment-specific parameters detailed in Table 2. We investigate two types of robustness:

Robustness to Agent Disturbance. We measure this by evaluating learned policies under a worst-case adversary, which minimizes the agent’s return without modifying the policy parameters.

Robustness to Environmental Change. We evaluate how well policies generalize to environments with variations in physical parameters such as mass and friction. Generalization is assessed by deploying policies in environments with shifted dynamics coefficients.

To analyze performance, we group MuJoCo tasks into three domains: cartpole, reacher, and locomotion (e.g., quadruped, cheetah, walker, hopper). For all experiments, both standard RL baselines (SAC and QARL) and ICRL methods are deployed in the same training environments used by SAC, favoring the former. *This gives SAC and QARL significant advantages* and explains why all ICRL methods other than ours are outperformed by QARL while *ICAG consistently outperforms QARL and other ICRL methods in all evaluations.*

Performance metrics are visualized through box plots and heatmaps in Figure 9-17. Note: PDT denotes prompt-DT (Xu et al., 2022), and MDPT refers to mix-DPT (Wang et al., 2024). Specifically, ICAG improves robustness against adversaries compared to DPT. Consistent with findings in standard robust RL (Pinto et al., 2017; Dong et al., 2025b), we observe that training for robustness can also

improve performance in disturbance-free settings. ICAG particularly excels in high-dimensional control problems, such as the quadruped task (Figure 11).

Furthermore, ICAG demonstrates strong generalization under out-of-distribution (OOD) environmental changes. In Figure 12-17, we evaluate how methods adapt when tested on parameter grids not seen during training. Each 8×8 heatmap varies two internal environment parameters, while the pretraining dataset only covers the central 4×4 grid. ICAG consistently achieves the best performance on these OOD environments, outperforming all baselines in tests such as Figure 15(b) and Figure 16(a)

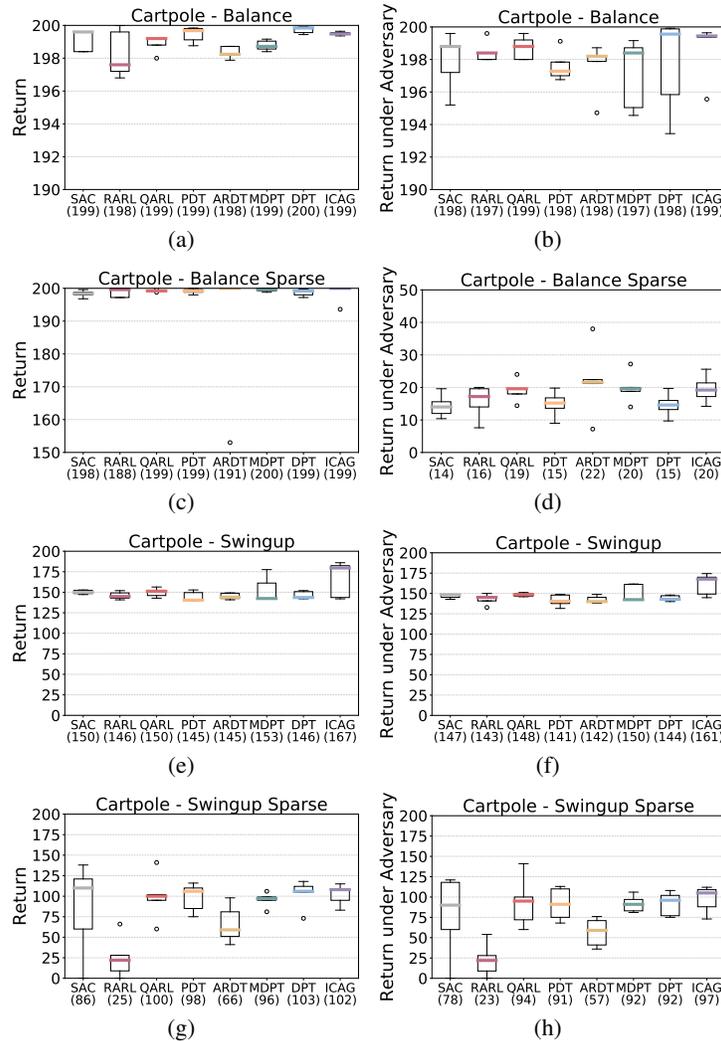


Figure 9: Performance and robustness on Cartpole problems (see the title of each boxplot), which are evaluated at the end of training without an adversary (left column) and against an adversary (right column). The number next to the name of each algorithm is the average performance across 10 seeds..

F THEORETICAL RESULTS

Here we provide theoretical guarantees for ICAG to gain further insights into their efficacy. Complete proofs of results in this section can be found in Appendix G.

We present two results. Our **first** result show that ICAG addresses the adversaries (disturbances) encountered during deployment in a manner similar to Posterior Sampling (PS) Osband & Van Roy (2017), which is widely recognized as the most sample-efficient algorithm for many sequential

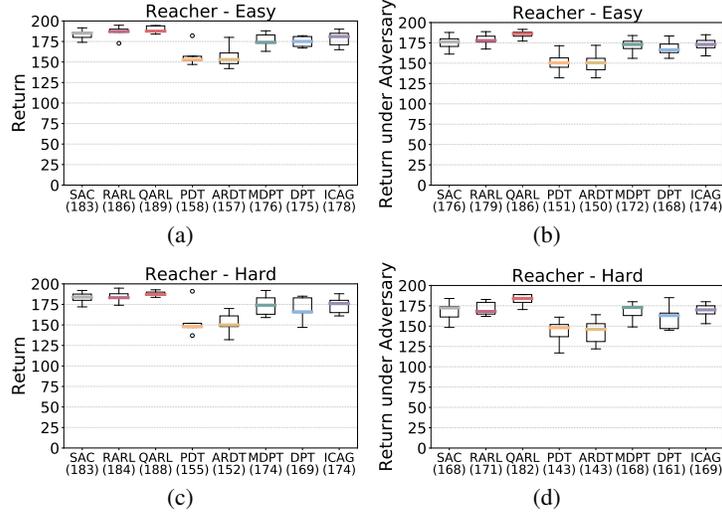


Figure 10: Performance and robustness on Reacher problems (see the title of each boxplot), which are evaluated at the end of training without an adversary (left column) and against an adversary (right column). The number next to the name of each algorithm is the average performance across 10 seeds.

decision-making problems. Our **second** result shows that ICAG continuously improve the quality of its action labels so that we can improve the performance of ICRL models in each ICAG iteration.

To facilitate analysis, we consider a slightly modified supervised pretraining framework similar to Lee et al. (2024) and Lin et al. (2024) for ICAG where, for any variation environment (τ, ϕ) , the TMs also condition on a sequence $\zeta_h = (s_1, a_1^*, s_2, \dots, s_h, a_h^*)$ where $s_{1:h}$ follows the distribution $p_s \in \Delta(S^h)$ and $a_j^* \sim \pi_{\tau, \phi}^*(s_h)$ where $\pi_{\tau, \phi}^*$ is the optimal policy for (τ, ϕ) as defined in Equation 5. Thus, the joint distribution over $(\tau, \phi, D, s_{\text{query}}, \zeta_h)$ for ICAG pretraining is

$$P(\tau, \phi, D, s_{\text{query}}, \zeta_h) = p_\tau(\tau) p_{\Phi(\tau)}(\phi) p_D(D; \tau, \phi) p_{\text{query}}(s_{\text{query}}) p_s(s_{1:h}) \prod_{j=1}^h \pi_{\tau, \phi}^*(a_j^* | s_h), \quad (6)$$

where p_τ and $p_{\Phi(\tau)}$ are the sampling distributions of the environment and disturbance for ICAG, respectively; $p_D(D; \tau, \phi)$ is the distribution of context dataset D given the variation environment (τ, ϕ) ; p_{query} is the distribution for sampling query states. Given the joint distribution in Equation 6, posterior distributions such as $P(\tau, \phi | D)$ are well-defined.

Consider the following general PS process for a fixed task τ' and disturbance ϕ' : initialize the posterior distribution as the ICRL pretraining distribution $p_{\tau, \phi}^{(1)} = p_\tau p_{\Phi(\tau)}$, and initialize an empty dataset D to collect transitions; for $h \in \{1, \dots, H\}$: **(i)** sample a variation environment $(\tau^{(h)}, \phi^{(h)}) \sim p_{\tau, \phi}^{(h)}$; **(ii)** solve for $\pi_{\tau^{(h)}, \phi^{(h)}}^*$; **(iii)** given the current state $s^{(h)}$, take an action following $a^{(h)} \sim \pi_{\tau^{(h)}, \phi^{(h)}}^*(s^{(h)})$, and observe the reward $r^{(h)}$ and next state $s^{(h+1)}$; **(iv)** add the transition $(s^{(h)}, a^{(h)}, r^{(h)}, s^{(h+1)})$ into D , and update the posterior $p_{\tau, \phi}^{(h+1)} = P(\tau, \phi | D)$.

Note that although PS is provably sample-efficient, computing the posterior $P(\tau, \phi | D)$ is often intractable in practice. Next, we prove that ICAG is an implicit PS: during deployment, ICAG takes actions like the PS process above, inferring the underlying environment and adversary without explicitly computing the posterior. We first make some common mild assumptions for analysis Lee et al. (2024).

Assumption F.1. Consider the context dataset $D = \{s_h, a_h, r_h, s_{h+1}\}_h$. The actions a_h are conditionally independent of the variation environment (τ, ϕ) given the history, i.e., $p_D(a_h | s_h, D_{h-1}) = p_D(a_h | s_h, D_{h-1}, \tau, \phi)$ where $D_h = \{s_{h'}, a_{h'}, r_{h'}, s_{h'+1}\}_{h' \leq h}$.

1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349

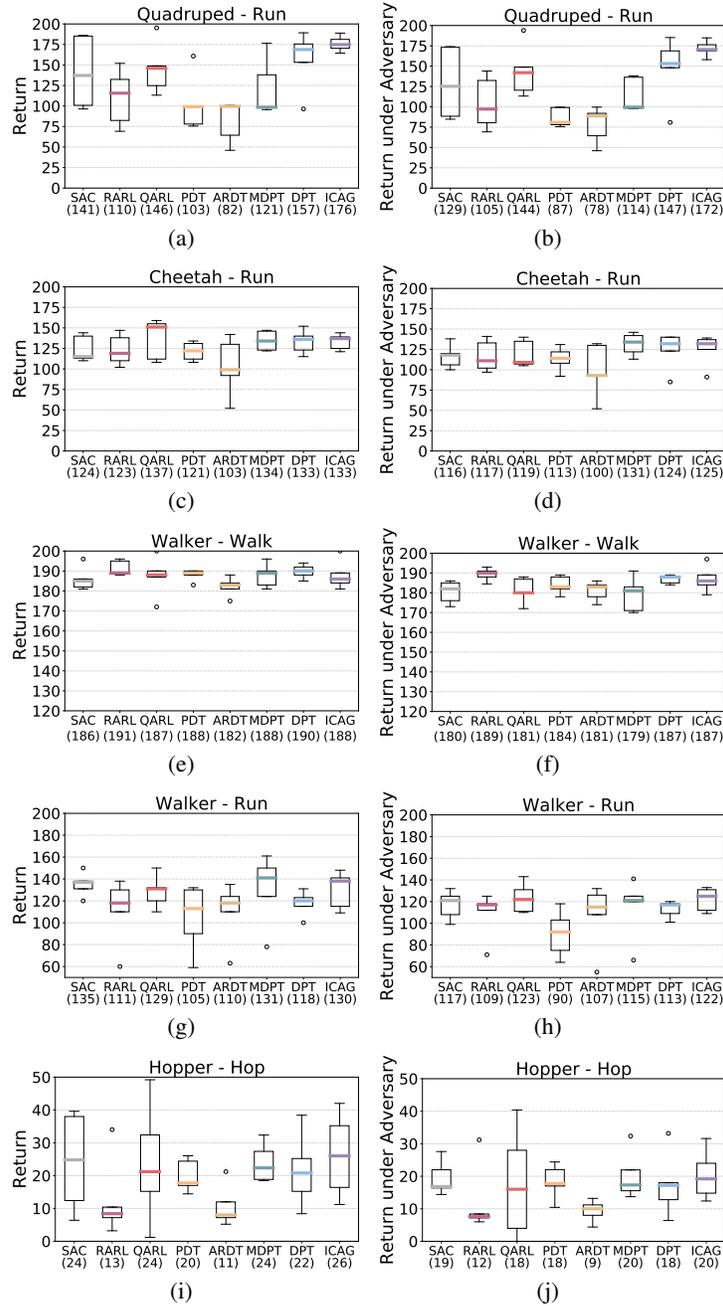


Figure 11: Performance and robustness on locomotion problems, i.e., Quadruped, Cheetah, Walker, and Hopper (see the title of each boxplot), which are evaluated at the end of training without an adversary (left column) and against an adversary (right column). The number next to the name of each algorithm is the average performance across 10 seeds.

1350

1351

1352

1353

1354

1355

1356

1357

1358

1359

1360

1361

1362

1363

1364

1365

1366

1367

1368

1369

1370

1371

1372

1373

1374

1375

1376

1377

1378

1379

1380

1381

1382

1383

1384

1385

1386

1387

1388

1389

1390

1391

1392

1393

1394

1395

1396

1397

1398

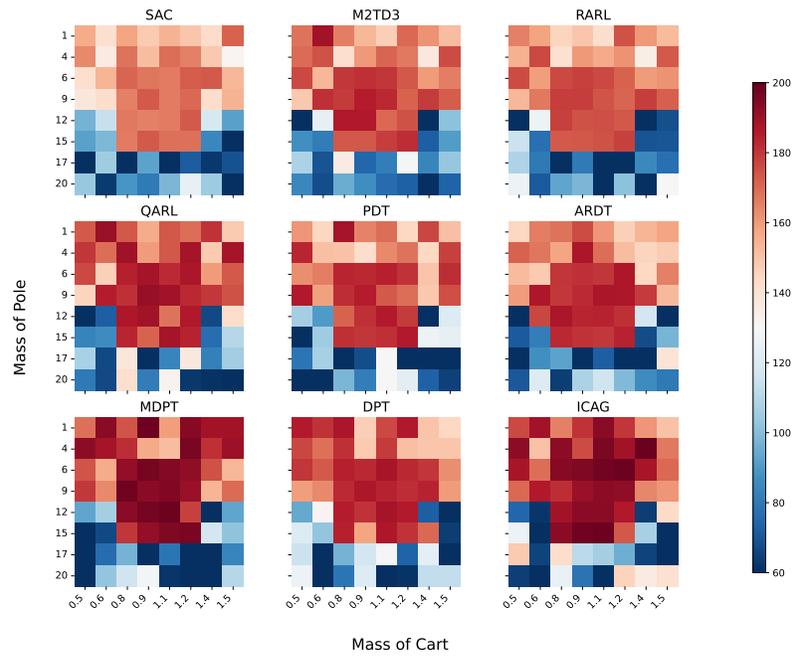
1399

1400

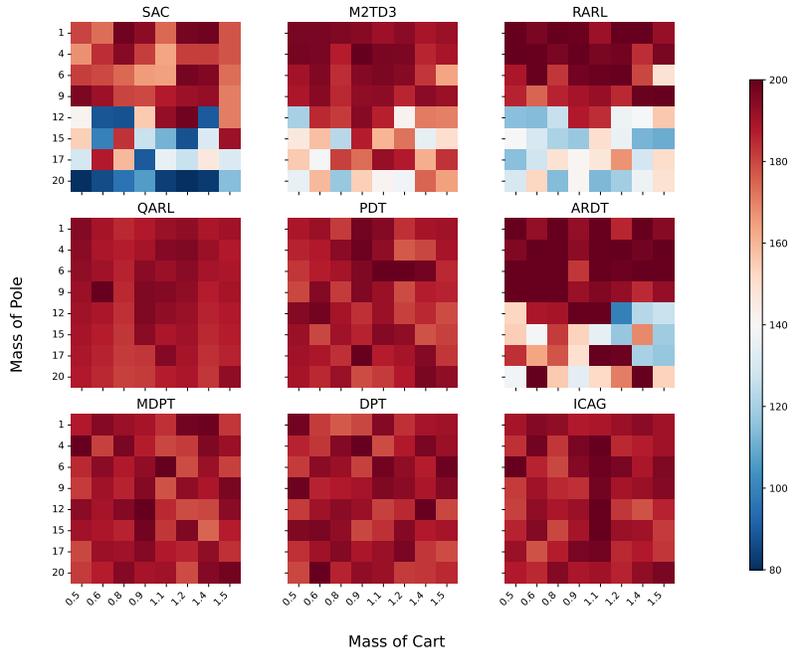
1401

1402

1403



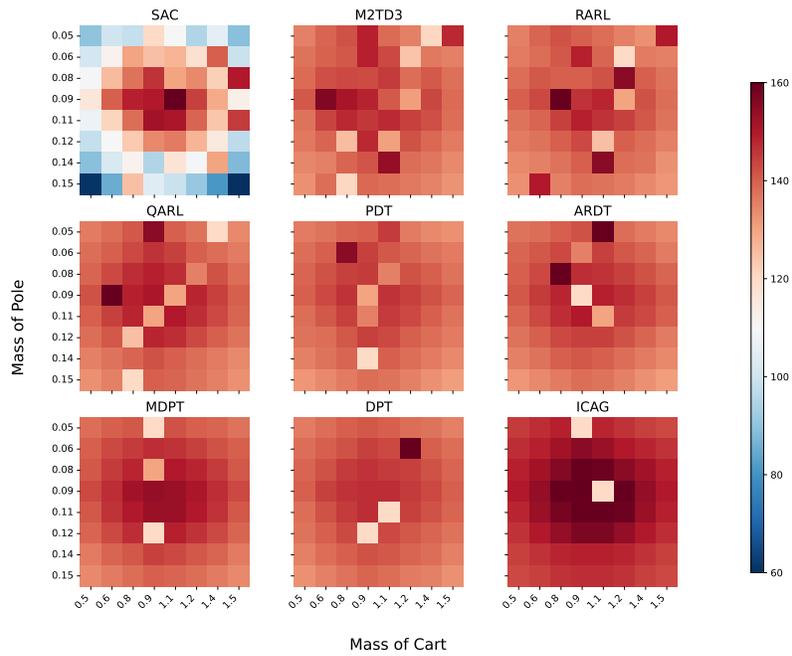
(a) Cartpole-Balance



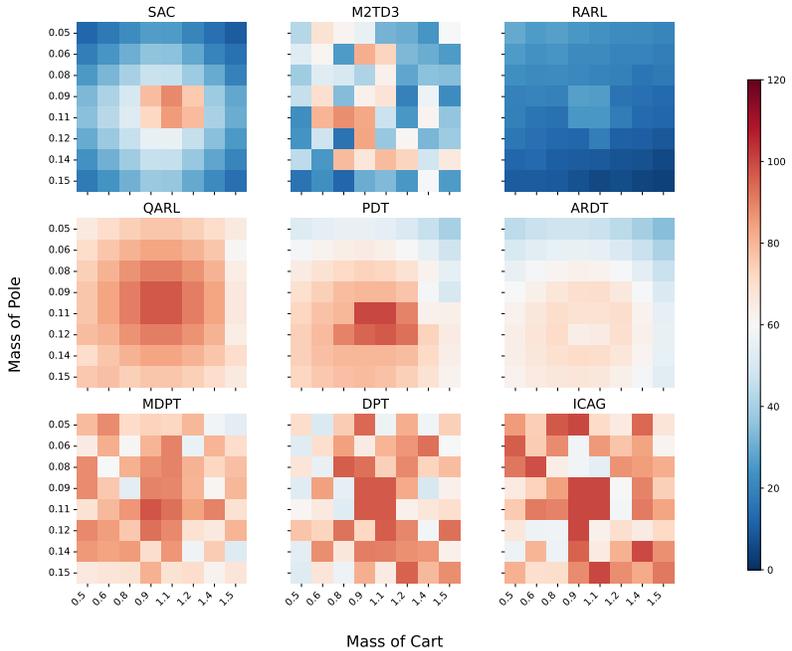
(b) Cartpole-Balance-Sparse

Figure 12: Robustness analysis on Cartpole Balance (see the title of each heatmap set). Heatmaps show the performance obtained after training for varying properties of the environment, described in the $x - y$ axes. The number next to the name of each algorithm is the average performance across 10 seeds.

1404
 1405
 1406
 1407
 1408
 1409
 1410
 1411
 1412
 1413
 1414
 1415
 1416
 1417
 1418
 1419
 1420
 1421
 1422
 1423
 1424
 1425
 1426
 1427
 1428
 1429
 1430
 1431
 1432
 1433
 1434
 1435
 1436
 1437
 1438
 1439
 1440
 1441
 1442
 1443
 1444
 1445
 1446
 1447
 1448
 1449
 1450
 1451
 1452
 1453
 1454
 1455
 1456
 1457



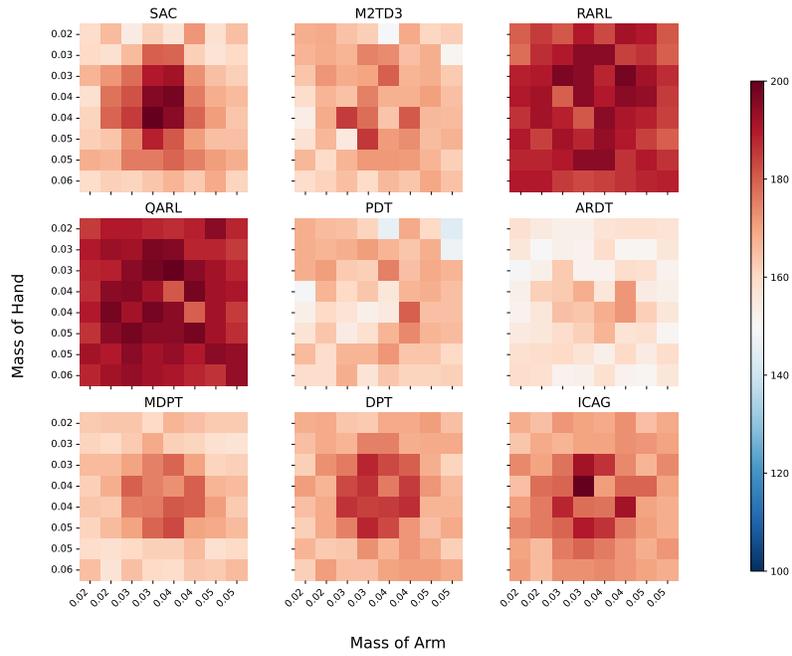
(a) Cartpole-Swingup



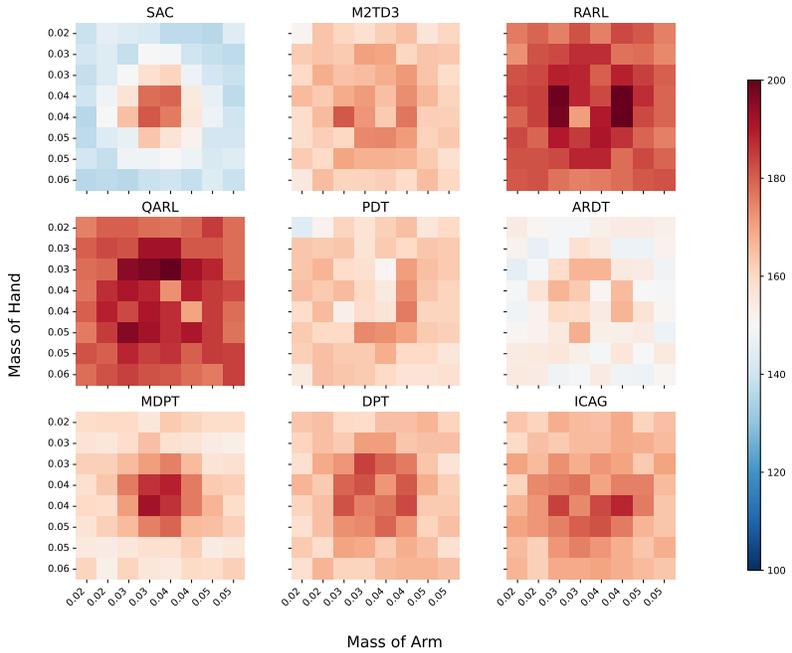
(b) Cartpole-Swingup-Sparse

Figure 13: Robustness analysis on Cartpole Swingup (see the title of each heatmap set). Heatmaps show the performance obtained after training for varying properties of the environment, described in the $x - y$ axes. The number next to the name of each algorithm is the average performance across 10 seeds.

1458
 1459
 1460
 1461
 1462
 1463
 1464
 1465
 1466
 1467
 1468
 1469
 1470
 1471
 1472
 1473
 1474
 1475
 1476
 1477
 1478
 1479
 1480
 1481
 1482
 1483
 1484
 1485
 1486
 1487
 1488
 1489
 1490
 1491
 1492
 1493
 1494
 1495
 1496
 1497
 1498
 1499
 1500
 1501
 1502
 1503
 1504
 1505
 1506
 1507
 1508
 1509
 1510
 1511



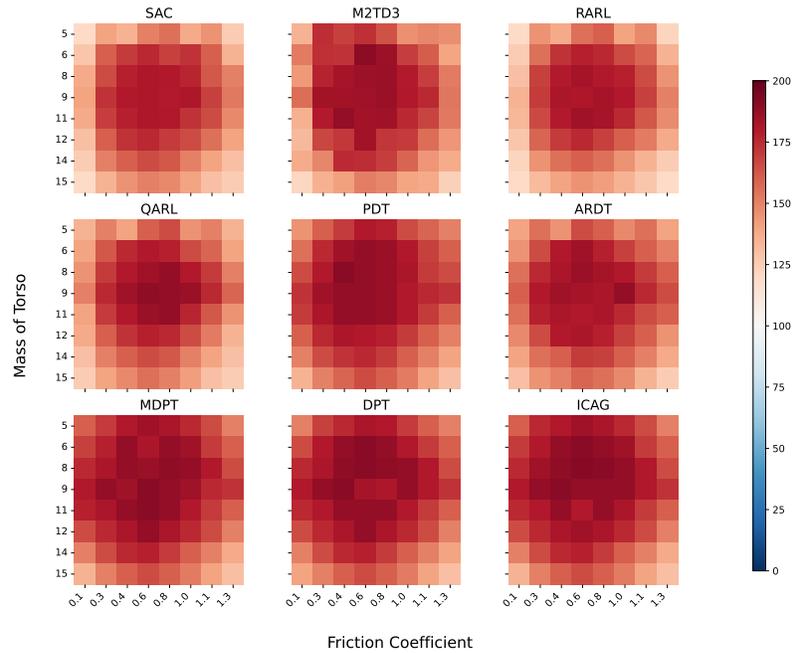
(a) Reacher-Easy



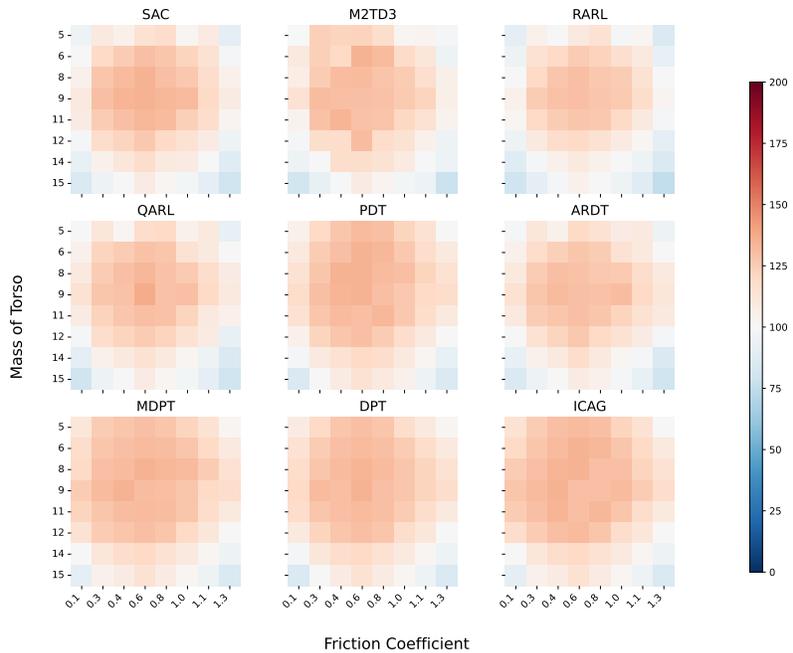
(b) Reacher-Hard

Figure 14: Robustness analysis on Reacher (see the title of each heatmap set). Each heatmap shows the performance obtained after training for varying properties of the environment, described in the $x - y$ axes. The number next to the name of each algorithm is the average performance across 10 seeds.

1512
 1513
 1514
 1515
 1516
 1517
 1518
 1519
 1520
 1521
 1522
 1523
 1524
 1525
 1526
 1527
 1528
 1529
 1530
 1531
 1532
 1533
 1534
 1535
 1536
 1537
 1538
 1539
 1540
 1541
 1542
 1543
 1544
 1545
 1546
 1547
 1548
 1549
 1550
 1551
 1552
 1553
 1554
 1555
 1556
 1557
 1558
 1559
 1560
 1561
 1562
 1563
 1564
 1565



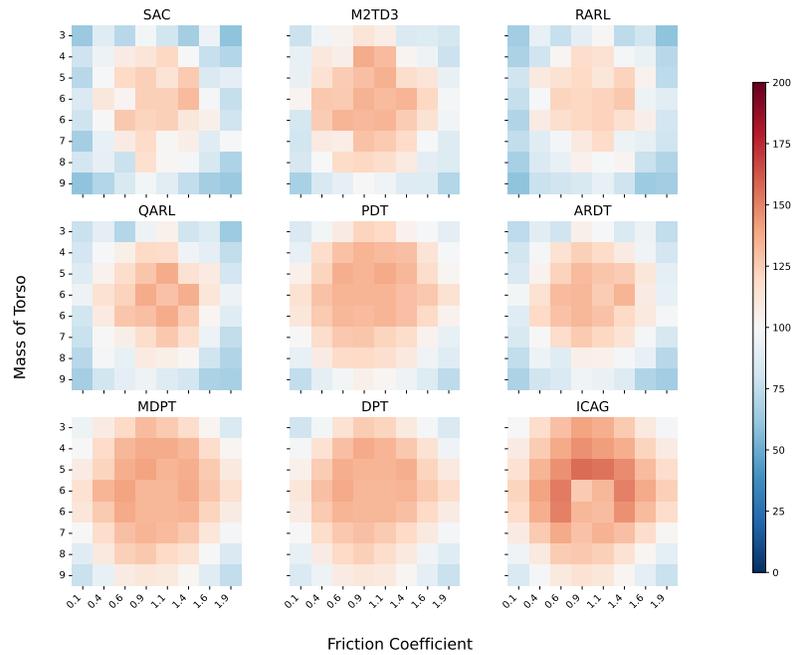
(a) Walker-Walk



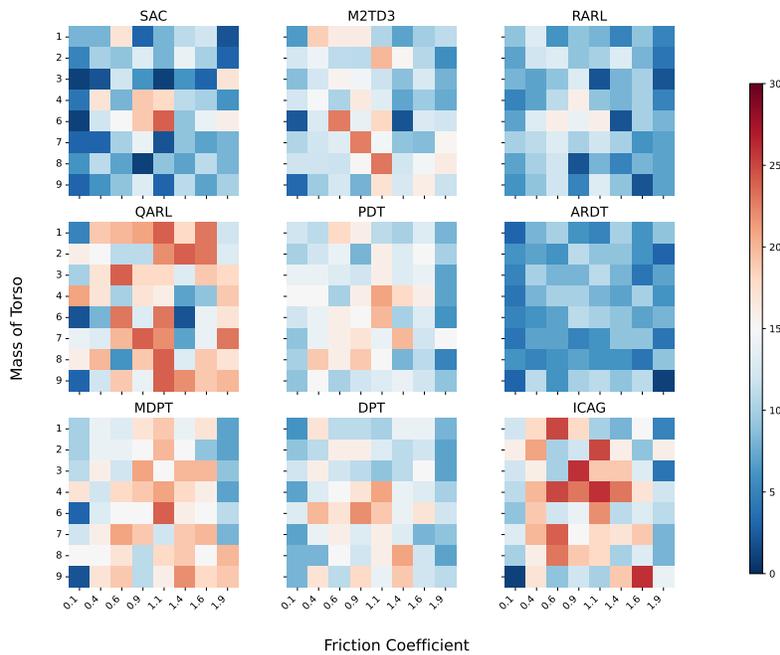
(b) Walker-Run

Figure 15: Robustness analysis on locomotion problems (Walker) (see the title of each heatmap set). Each heatmap shows the performance obtained after training for varying properties of the environment, described in the $x - y$ axes. The number next to the name of each algorithm is the average performance across 10 seeds.

1566
 1567
 1568
 1569
 1570
 1571
 1572
 1573
 1574
 1575
 1576
 1577
 1578
 1579
 1580
 1581
 1582
 1583
 1584
 1585
 1586
 1587
 1588
 1589
 1590
 1591
 1592
 1593
 1594
 1595
 1596
 1597
 1598
 1599
 1600
 1601
 1602
 1603
 1604
 1605
 1606
 1607
 1608
 1609
 1610
 1611
 1612
 1613
 1614
 1615
 1616
 1617
 1618
 1619

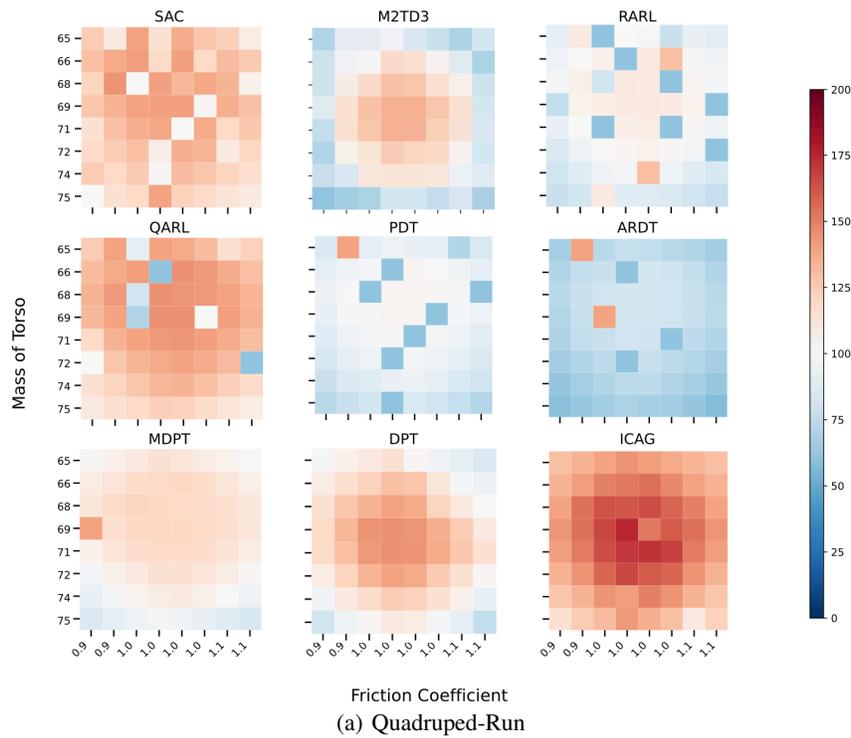


(a) Cheetah-Run



(b) Hopper-Hop

Figure 16: Robustness analysis on locomotion problems, i.e., Cheetah and Hopper (see the title of each heatmap set). Each heatmap shows the performance obtained after training for varying properties of the environment, described in the $x - y$ axes. The number next to the name of each algorithm is the average performance across 10 seeds.



1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673

Figure 17: Robustness analysis on locomotion problems (Quadruped) (see the title of each heatmap set). Each heatmap shows the performance obtained after training for varying properties of the environment, described in the $x - y$ axes. The number next to the name of each algorithm is the average performance across 10 seeds.

This assumption in essence assumes that the behavioral policies for collecting context datasets are functions of the history only. This holds, for example, when the context dataset is collected by random policies not depending on current states or any RL algorithms only using the history.

Assumption F.2. Consider a sufficiently expressive and pretrained TM T_θ . For all $(s_{\text{query}}, D, \zeta_h)$, $T_\theta(a|s_{\text{query}}, D, \zeta_h) = P(a|s_{\text{query}}, D, \zeta_h)$ for all a .

The purpose of Assumption F.2, which states that the pretrained TM T_θ matches the pretraining distribution P , is to focus the analysis on ICRL deployment rather than the quality of pretraining. This assumption is a common assumption for ICRL Lee et al. (2024) and in-context learning analysis Xie et al. (2022). To see why this assumption is valid, it is well-established that deep learning models, such as T_θ , are universal approximators Scarselli & Tsoi (1998). Moreover, the maximum likelihood (ML)-based pretraining loss for T_θ is equivalent to find a minimizer of the following expected Kullback–Leibler (KL) divergence

$$\mathbb{E}_{(s_{\text{query}}, D, \zeta_h) \sim P} [\text{KL}(P(a|s_{\text{query}}, D, \zeta_h) \| T_\theta(a|s_{\text{query}}, D, \zeta_h))],$$

where $\text{KL}(p, q) = \mathbb{E}_p[\log(p/q)]$ is the KL divergence between two distributions. Assuming sufficient expressiveness, the above divergence can be minimized at $T_\theta = P$. Thus, with extra coverage assumptions regarding P and sufficient amount of pretraining data, Assumption F.2 can hold with T_θ and P arbitrarily close to each other. However, we omit this proof as this would distract the focus of the analysis. Next, we present our main theoretical results.

Theorem F.3. Fix an environment τ' with a disturbance ϕ' for deployment and a context dataset $D \sim p_D(D; \tau', \phi')$ for the pretrained TM T_θ to condition on for ICRL. Consider the random sequence $\Upsilon_h = (S^{(1)}, A^{(1)}, S^{(2)}, A^{(2)}, \dots, S^{(h)}, A^{(h)})$. It holds that

$$P_{PS}(\Upsilon_h | \tau', \phi', D) = P_\theta(\Upsilon_h | \tau', \phi', D),$$

where (i) $P_{PS}(\Upsilon_h | \tau', \phi', D)$ is the distribution of Υ_h following the PS algorithm: $(\tau_{ps}, \phi_{ps}) \sim P(\tau, \phi | D)$, $S^{(1)} \sim \rho_{\tau', \phi'}$, for all $h' \leq h$, $A^{(h')} \sim \pi_{\tau_{ps}, \phi_{ps}}^*(S^{(h')})$, $S^{h'+1} \sim P_{\tau', \phi'}(S^{(h)}, A^{(h)})$; (ii) $P_\theta(\Upsilon_h | \tau', \phi', D)$ is the distribution of Υ_h following ICRL with T_θ : $S^{(1)} \sim \rho_{\tau', \phi'}$, for all $h' \leq h$, $A^{(h')} \sim T_\theta(a|S^{(h')}, D, \Upsilon_{h'-1})$, and $S^{h'+1} \sim P_{\tau', \phi'}(S^{(h')}, A^{(h')})$.

Theorem F.3 states that the trajectory distribution under ICRL with the pretrained TM T_θ is the same as the trajectory distribution under PS, establishing that ICAG pretrains TMs for implicit PS. In particular, ICAG implicitly estimates the posterior distribution of the environment τ and the adversary ϕ so that ICAG can act optimally if there exists an adversary ϕ that can perturb the environment τ . Moreover, this process of estimating and adapting to a potential adversary is provably sample-efficient given the optimal sample efficiency of PS.

Next, we show that ICAA can refine its action labels in an iterative manner.

Assumption F.4 (Posterior Consistency). For any pair of underlying task and adversary τ^*, ϕ^* , as the context dataset D contains more transitions, the posterior $P(\tau, \phi | D)$ concentrates toward the true task and adversary, i.e., for any neighbor U of (τ^*, ϕ^*) , it holds that $P(U | D) \rightarrow 1$ as $|D| \rightarrow +\infty$.

Assumption F.4 is a standard assumption and, in general, a fact for any Bayesian method. Following Theorem F.3 that ICRL models are performing Posterior Sampling, this implies that the transformer policy T_θ has increasing performance when the context size $|D|$ increases.

Theorem F.5. Under Assumptions F.1, F.2 and F.4, in every iteration of ICAA (Algorithm 2), and for every variation environment (τ, ϕ) with sufficient exploration, the action label generation policy achieves performance no worse than the transformer policy T_θ from the previous iteration within the same environment (τ, ϕ) .

In particular, by matching the action labels generated by a stronger policy for the variation environment (τ, ϕ) in the finetuning stage of each ICAA iterations, the performance of transformer policy T_θ increases and can generalize better to new tasks and adversaries. Theorem F.5 implies that ICAA continues to improve the quality of its action labels until it saturates.

G PROOFS OF THEORETICAL RESULTS

G.1 PROOF OF THEOREM F.3

Fix an environment τ' with a disturbance ϕ' for deployment and a context dataset $D \sim p_D(D; \tau', \phi')$ for the pretrained TM T_θ to condition on for ICRL. Consider the random sequence $\Upsilon_h = (S^{(1)}, A^{(1)}, S^{(2)}, A^{(2)}, \dots, S^{(h)}, A^{(h)})$. It holds that

$$P_{PS}(\Upsilon_h | \tau', \phi', D) = P_\theta(\Upsilon_h | \tau', \phi', D),$$

where (i) $P_{PS}(\Upsilon_h | \tau', \phi', D)$ is the distribution of Υ_h following the PS algorithm: $(\tau_{ps}, \phi_{ps}) \sim P(\tau, \phi | D)$, $S^{(1)} \sim \rho_{\tau', \phi'}$, for all $h' \leq h$, $A^{(h')} \sim \pi_{\tau_{ps}, \phi_{ps}}^*(S^{(h')})$, $S^{h'+1} \sim P_{\tau', \phi'}(S^{(h')}, A^{(h')})$; (ii) $P_\theta(\Upsilon_h | \tau', \phi', D)$ is the distribution of Υ_h following ICRL with T_θ : $S^{(1)} \sim \rho_{\tau', \phi'}$, for all $h' \leq h$, $A^{(h')} \sim T_\theta(a | S^{(h')}, D, \Upsilon_{h'-1})$, and $S^{h'+1} \sim P_{\tau', \phi'}(S^{(h')}, A^{(h')})$.

Proof of Theorem F.3. The proof is based on induction on h . Given that the results are for any fixed τ', ϕ', D , we omit the conditional dependence on them to improve clarity when there is no confusion. Recall that P denotes the ICAG pretraining dataset distribution as defined in Equation 6. We first prove a result to be used in the proof. Under Assumption F.1, we have

$$P_{PS}(\tau_{ps} = ps, \phi_{ps} = \phi | D) = P(\tau, \phi | D), \quad (7)$$

where $P(\tau | D)$ is the posterior distribution of the ICAG pretraining distribution P . This follows from

$$\begin{aligned} P_{PS}(\tau_{ps} = \tau, \phi_{ps} = \phi | D) &\propto P_{PS}(\tau_{ps}, \phi_{ps} = \phi, D) = P(\tau, \phi) P_{PS}(D | \tau, \phi) \\ &\propto P(\tau, \phi) \rho_{\tau', \phi'}(s_1) \prod_{(s_h, a_h, r_h, s_{h+1}) \in D} P_{\tau', \phi'}(s_{h+1} | s_h, a_h) p_D(a_h | D_{h-1}) \\ &\propto P(\tau, \phi) P(D; \tau; \phi) = P(\tau, \phi, D) \propto P(\tau, \phi | D), \end{aligned}$$

where the second \propto is due to Assumption F.1 so that we can plug in $p_D(a_h | D_{h-1})$. Now we begin the proof of induction.

Case $h = 1$. We have

$$\begin{aligned} P_{PS}(S^{(1)}, A^{(1)}) &= P_{PS}(S^{(1)}) P_{PS}(A^{(1)} | S^{(1)}) = p_{\tau', \phi'}(S^{(1)}) \int_{\tau, \phi} P_{PS}(A^{(1)}, \tau_{ps} = \tau, \phi_{ps} = \phi | S^{(1)}) d\tau d\phi \\ &= p_{\tau', \phi'}(S^{(1)}) \int_{\tau} P_{PS}(A^{(1)} | S^{(1)}, \tau_{ps} = \tau, \phi_{ps} = \phi) P_{PS}(\tau_{ps} = \tau, \phi_{ps} = \phi | S^{(1)}) d\tau d\phi \\ &= p_{\tau', \phi'}(S^{(1)}) \int_{\tau} \pi_{\tau, \phi}^*(A^{(1)} | S^{(1)}) P_{PS}(\tau_{ps} = \tau, \phi_{ps} = \phi | S^{(1)}) d\tau d\phi. \end{aligned}$$

To continue, note that $P_{PS}(\tau_{ps} = \tau, \phi_{ps} = \phi | S^{(1)}) = P_{PS}(\tau_{ps} = \tau, \phi_{ps} = \phi | D)$ given that the posterior sampling does not depend on $S^{(1)}$. With Equation 7, we further have

$$P_{PS}(\tau_{ps} = \tau, \phi_{ps} = \phi | D) = P(\tau, \phi | D).$$

Thus,

$$\begin{aligned} P_{PS}(S^{(1)}, A^{(1)}) &= p_{\tau', \phi'}(S^{(1)}) \int_{\tau, \phi} \pi_{\tau, \phi}^*(A^{(1)} | S^{(1)}) P(\tau, \phi | D) d\tau d\phi \\ &= p_{\tau', \phi'}(S^{(1)}) \int_{\tau, \phi} P(A^{(1)} | \tau, \phi, S^{(1)}) P(\tau, \phi | D) d\tau d\phi \\ &= p_{\tau', \phi'}(S^{(1)}) P(A^{(1)} | S^{(1)}) = p_{\tau', \phi'}(S^{(1)}) P_\theta(A^{(1)} | S^{(1)}) = P_\theta(S^{(1)}, A^{(1)}), \end{aligned}$$

where the last line is due to and $T_\theta(A^{(1)} | S^{(1)}, D) = P(A^{(1)} | S^{(1)}, D)$ under Assumption F.2.

Case h . Assume that Case $h - 1$ holds that

$$P_{PS}(\Upsilon_{h-1}) = P_\theta(\Upsilon_{h-1}),$$

and we aim to prove $P_{PS}(\Upsilon_h) = P_\theta(\Upsilon_h)$, which is equivalent to prove

$$P_{PS}(S^{(h)}, A^{(h)} | \Upsilon_{h-1}) = P_\theta(S^{(h)}, A^{(h)} | \Upsilon_{h-1}),$$

1782 because of the factorization

$$1783 P_{PS}(S^{(h)}, A^{(h)} | \Upsilon_{h-1}) P_{PS}(\Upsilon_{h-1}) = P_\theta(S^{(h)}, A^{(h)} | \Upsilon_{h-1}) P_\theta(\Upsilon_{h-1}).$$

1785 To this end, we have

$$1786 P_{PS}(S^{(h)}, A^{(h)} | \Upsilon_{h-1}) = P_{PS}(S^{(h)} | \Upsilon_{h-1}) P_{PS}(A^{(h)} | S^{(h)}, \Upsilon_{h-1}) = P_{\tau', \phi'}(S^{(h)} | \Upsilon_{h-1}) P_{PS}(A^{(h)} | S^{(h)}, \Upsilon_{h-1})$$

$$1787 = P_{\tau', \phi'}(S^{(h)} | \Upsilon_{h-1}) \int_{\tau, \phi} P_{PS}(A^{(h)}, \tau_{ps} = \tau, \phi_{ps} = \phi | S^{(h)}, \Upsilon_{h-1}) d\tau d\phi$$

$$1788 = P_{\tau', \phi'}(S^{(h)} | \Upsilon_{h-1}) \int_{\tau, \phi} P_{PS}(A^{(h)}, \tau_{ps} = \tau, \phi_{ps} = \phi | S^{(h)}, \Upsilon_{h-1}) d\tau d\phi$$

$$1789 = P_{\tau', \phi'}(S^{(h)} | \Upsilon_{h-1}) \int_{\tau, \phi} P_{PS}(A^{(h)} | \tau_{ps} = \tau, \phi_{ps} = \phi, S^{(h)}, \Upsilon_{h-1}) P_{PS}(\tau_{ps} = \tau, \phi_{ps} = \phi | S^{(h)}, \Upsilon_{h-1}) d\tau d\phi$$

$$1790 = P_{\tau', \phi'}(S^{(h)} | \Upsilon_{h-1}) \int_{\tau, \phi} \pi_{\tau, \phi}^*(A^{(1)} | S^{(1)}) P_{PS}(\tau_{ps} = \tau, \phi_{ps} = \phi | S^{(h)}, \Upsilon_{h-1}) d\tau d\phi.$$

1792 To continue, we prove that $P_{PS}(\tau_{ps} = \tau, \phi_{ps} = \phi | S^{(h)}, \Upsilon_{h-1}) = P(\tau, \phi | S^{(h)}, \Upsilon_{h-1})$. Indeed,

$$1793 P_{PS}(\tau, \phi | S^{(h)}, \Upsilon_{h-1}) = P_{PS}(\tau, \phi, S^{(h)}, \Upsilon_{h-1}) / P_{PS}(S^{(h)}, \Upsilon_{h-1})$$

$$1800 \propto P_{PS}(\tau, \phi, S^{(h)}, \Upsilon_{h-1})$$

$$1801 = P_{PS}(\Upsilon_{h-1} | \tau, \phi) P_{PS}(S^{(h)} | \Upsilon_{h-1}) P_{PS}(\tau, \phi | D)$$

$$1802 \propto P(\tau, \phi | D) p_s(S^{(1:h)}) \prod_{h' \leq h} \pi_{\tau, \phi}^*(A^{(h')} | S^{(h')})$$

$$1803 = P(\tau, \phi, S^{(h)}, \Upsilon_{h-1}) \propto P(\tau, \phi | S^{(h)}, \Upsilon_{h-1}).$$

1804 Given that $P_{PS}(\tau, \phi | S^{(h)}, \Upsilon_{h-1})$ and $P(\tau, \phi | S^{(h)}, \Upsilon_{h-1})$ are distributions,

1805 $P_{PS}(\tau, \phi | S^{(h)}, \Upsilon_{h-1}) \propto P(\tau, \phi | S^{(h)}, \Upsilon_{h-1})$ implies that $P_{PS}(\tau, \phi | S^{(h)}, \Upsilon_{h-1}) =$

1806 $P(\tau, \phi | S^{(h)}, \Upsilon_{h-1})$. Thus,

$$1807 P_{PS}(S^{(h)}, A^{(h)} | \Upsilon_{h-1}) = P_{\tau', \phi'}(S^{(h)} | \Upsilon_{h-1}) \int_{\tau, \phi} \pi_{\tau, \phi}^*(A^{(h)} | S^{(h)}) P(\tau, \phi | S^{(h)}, \Upsilon_{h-1}) d\tau d\phi$$

$$1808 = P_{\tau', \phi'}(S^{(h)} | \Upsilon_{h-1}) \int_{\tau, \phi} P(A^{(h)} | \tau, \phi, S^{(h)}, \Upsilon_{h-1}) P(\tau, \phi | S^{(h)}, \Upsilon_{h-1}) d\tau d\phi$$

$$1809 = P_{\tau', \phi'}(S^{(h)} | \Upsilon_{h-1}) P(A^{(h)} | S^{(h)}, \Upsilon_{h-1})$$

$$1810 = P_{\tau', \phi'}(S^{(h)} | \Upsilon_{h-1}) P_\theta(A^{(h)} | S^{(h)}, \Upsilon_{h-1}) = P_\theta(S^{(h)}, A^{(h)} | \Upsilon_{h-1}),$$

1811 where the last line is due to Assumption F.2. Hence, the induction is complete and this concludes the

1812 proof. \square

1822 G.2 PROOF OF THEOREM F.5

1823 Under Assumptions F.1, F.2 and F.4, in every iteration of ICAA (Algorithm 2), and for every varia-

1824 tion environment (τ, ϕ) with sufficient exploration $|D|$, the action label generation policy achieves

1825 performance no worse than the transformer policy T_θ from the previous iteration within the same

1826 environment (τ, ϕ) .

1827 *Proof of Theorem F.5.* We first prove a useful lemma.

1828 **Lemma G.1.** Under Assumptions F.4, when deploying a pretrained T_θ to a task τ^* with an adversary

1829 ϕ^* , the performance of T_θ converges to that of the optimal policy, i.e.,

$$1830 \mathbb{E}[\mathcal{R}_{\tau^*}(T_\theta(\cdot | \cdot, D), \phi^*)] \rightarrow \max_{\pi} \mathcal{R}_{\tau^*}(\pi, \phi^*) \quad \text{as } |D| \rightarrow +\infty.$$

1831 *Proof of Lemma G.1.* As $|D| \rightarrow +\infty$, from Assumption F.4, we have the posterior $P(\tau, \phi | D)$ con-

1832 centrates toward the truth (τ^*, ϕ^*) . As proved by Theorem F.3, a pretrained T_θ is performance

Posterior Sampling during deployment. This leads to, for any neighbor $U(\epsilon)$ of (τ^*, ϕ^*) with radius $\epsilon > 0$,

$$\mathbb{E} [\mathcal{R}_{\tau^*} (T_\theta(\cdot|\cdot, D), \phi^*)] = \int P(\tau, \phi|D) \mathcal{R}_{\tau^*}(\pi_{\tau, \phi}^*, \phi^*) \quad (8)$$

$$\geq P(U|D) \inf_{\tau', \phi' \in U(\epsilon)} \mathcal{R}_{\tau^*}(\pi_{\tau', \phi'}^*, \phi^*) \xrightarrow{|D| \rightarrow +\infty, \epsilon \rightarrow 0} \mathcal{R}_{\tau^*}(\pi_{\tau^*, \phi^*}^*, \phi^*) = \max_{\tau} \mathcal{R}_{\tau^*}(\pi, \phi^*). \quad (9)$$

In addition, by definition of \mathcal{R} . we have $\mathbb{E} [\mathcal{R}_{\tau^*} (T_\theta(\cdot|\cdot, D), \phi^*)] \leq \max_{\tau} \mathcal{R}_{\tau^*}(\pi, \phi^*)$ almost surely. This proves that $\mathbb{E} [\mathcal{R}_{\tau^*} (T_\theta(\cdot|\cdot, D), \phi^*)] \rightarrow \max_{\tau} \mathcal{R}_{\tau^*}(\pi, \phi^*)$. \square

For any D of finite transitions, with Lemma G.1, we can outperform $T_\theta(\cdot|\cdot, D)$ by extending the exploration to have D' where $|D'| > |D|$ such that the expected performance of $T_\theta(\cdot|\cdot, D')$ is arbitrarily close to the optimal one. Here, by definition of ICAA algorithm, $T_\theta(\cdot|\cdot, D')$ is the action label generation policy, thus concluding the proof. \square

H PSEUDOCODE

Algorithm 3 Deployment of ICRL Models

```

1: Input: Pretrained transformer Model  $T_\theta$ ; Horizon of episodes  $H$ ; Number of episodes  $N$  for
   online testing; Offline dataset  $D_{\text{off}} = \{(s_h, a_h, s_{h+1}, r_h)\}_h$ , consisting of transitions collected
   by a behavioral policy.
2: // Offline Testing
3: for every time step  $h \in \{1, \dots, H\}$  do
4:   Observe state  $s_h$ 
5:   Sample action with  $T_\theta$ :
           
$$a_h \sim T_\theta(\cdot|s_h, D_{\text{off}})$$

6:   Collect reward  $r_h$ 
7: end for
8: // Online Testing
9: Initialize an empty online data buffer  $D_{\text{on}} = \{\}$ 
10: for every online trial  $n \in \{1, \dots, N\}$  do
11:   for every time step  $h \in \{1, \dots, H\}$  do
12:     Observe state  $s_h$ 
13:     Sample action with  $T_\theta$ :
           
$$a_h \sim T_\theta(\cdot|s_h, D_{\text{on}})$$

14:     Collect reward  $r_h$ 
15:   end for
16:   Append the collected transitions  $\{(s_h, a_h, s_{h+1}, r_h)\}_h$  into  $D_{\text{on}}$ 
17: end for

```

I LIMITATIONS

While ICAG involves generating expert policies across perturbed task variants, this step is performed efficiently in parallel and amortized over pretraining, making it feasible for a modest number of variants. ICAA uses self-generated labels derived from a robust pretrained model, which, despite not being optimal, have been shown empirically to improve performance with minimal data. These design choices strike a practical balance between robustness and scalability. While the proposed methods improve robustness under structured perturbations, their generalization is currently focused on tasks with in-distribution environment variations. Extending this framework to handle broader generalization—such as out-of-distribution task structures and disturbances—remains a valuable direction for future work.