ASYMMETRIC ON-POLICY DISTILLATION LEARNS IN-CONTEXT EXPLORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sequential decision-making under partial observability is difficult because agents must explore to infer hidden, task-relevant information before acting effectively. Learning such information-gathering behaviors *tabula rasa* via online reinforcement learning is sample-inefficient, whereas offline imitation fails to learn exploratory behaviors due to insufficient coverage of test-time histories. We study asymmetric student-teacher learning in partially observed settings. Direct imitation of privileged experts, however, fails because such experts never need to explore. In this work, we propose ASTEROID, a simple and effective online framework for learning test-time exploration. Our key idea is that history-conditioned student policies can learn to explore through online asymmetric distillation: the student first collects on-policy context under partial observability, and the privileged expert provides action labels conditioned on that context. Iteratively, as this context grows, the student learns to use past observations to infer hidden task information and act according to the inferred latent state, yielding exploration similar to efficient Bayesian posterior sampling. Across a diverse range of eleven different environments, including discrete exploration benchmarks, robotic manipulation, navigation, and Procgen environments, ASTEROID achieves $\sim 2\times$ higher returns under fixed exploration budgets and uses up to $\sim 100\times$ fewer interactions than baselines. Finally, our method transfers policies trained in simulation to vision-denied real-world robot manipulation, exhibiting efficient exploration and increasing success by $+40\%$ over standard distillation. [1]

1 INTRODUCTION

Sequential decision-making under partial observability is a central challenge in reinforcement learning (RL) (Kaelbling et al., 1998; Osband et al., 2013). In the real world, from robotics to embodied AI, agents receive only partial, noisy observations of the environment. Effective behavior therefore requires exploring to infer hidden information before acting (Igl et al., 2018; Zintgraf et al., 2021; Lee et al., 2023; Weihs et al., 2021; Wan et al., 2025). For example, a household robot searching for an object must use onboard sensing to infer likely object locations, track locations it has already searched, and adapt future exploration based on prior attempts (Chaplot et al., 2020b;a). Such problems are formalized as partially observable Markov decision processes (POMDPs) (Kaelbling et al., 1998).

Solving POMDPs requires policies with memory (Hausknecht and Stone, 2015; Ni et al., 2021). Learning such policies from scratch via meta-RL is sample-inefficient, since the agent must learn to gather information and exploit it at the same time (Bauer et al., 2023; Zintgraf et al., 2021; Duan et al., 2016). A way to improve sample efficiency is to assume training-time access to a "clairvoyant" expert that observes the hidden state and acts accordingly. This expert can teach the student how to act if the hidden state is known, but not how to acquire that information (Liu et al., 2021). Naively distilling its behavior therefore fails to learn policies that can explore under partial observability (Walsman et al., 2023). Interactive imitation (Swamy et al., 2022) offers a natural fix by relabeling student mistakes with expert actions. However, standard approaches (Ross et al., 2011) yield students that learn the expert's average action across histories that alias multiple hidden states and dither between hypotheses rather than resolving them (Kim et al., 2025; Walsman et al., 2023). Other approaches

[1] Project webpage: sites.google.com/view/icl-exploration

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

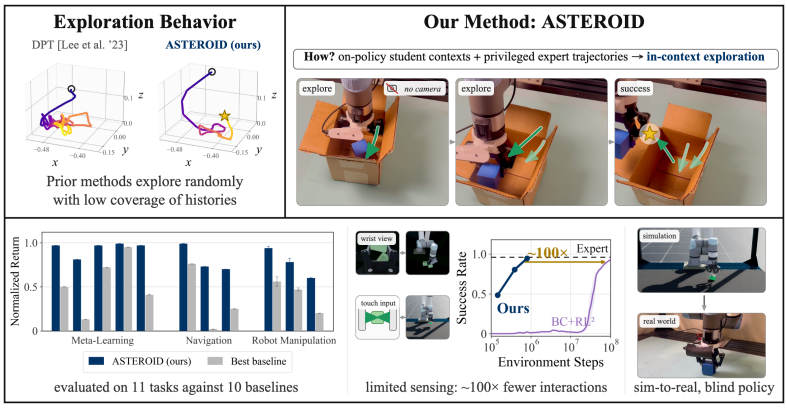


Figure 1: ASTEROID provides a simple and reliable method for learning efficient exploration in partially observed domains via on-policy asymmetric student-teacher distillation.

propose to address the student-teacher mismatch using expensive online RL (Weihs et al., 2021), differentiable experts (Warrington et al., 2021), or offline coverage assumptions (Lee et al., 2023).

In this work, we propose ASTEROID (**A**symmetric **S**tudent **Te**acher **R**oll-**O**ut **I**n-context **D**istillation), a framework for learning efficient exploration from interactive imitation of privileged experts. The key idea is to decouple the source of history and supervision. ASTEROID rolls out the current student under partial observability, producing a history of observations, actions, and rewards, then queries the expert for actions from the state reached after that history (similar to multi-step action supervision in Kelly et al. (2019)). The expert does not explore; it only labels the student histories. As a result, failed searches and probes become useful context: they specify evidence accumulated so far, while the expert label specifies the next actions given the hidden state.

Why does this teach exploration? A partial history represents uncertainty over the hidden state. Returning to household navigation: suppose the object could be in location A, B, or C. An episode is successful when the object is found, so a history that continues past `search(A)` implies A was empty. If the student searches A and finds nothing, that history rules out A but remains consistent with B and C. A privileged expert would label this history with `search(B)` when the object is at B, and `search(C)` when it is at C. The student therefore learns to act over the remaining plausible actions: if it fails at A, it samples `search(B)` or `search(C)`. But if it still fails at B, all expert labels are `search(C)` as that's the only remaining goal. Unlike random sampling of locations, the policy conditions on its history and never retries hypotheses it has ruled out. Across many training tasks, ambiguous histories are labelled with a distribution of potential expert trajectories. Learned policies sample from this distribution to explore, and the resulting feedback updates the history and eliminates additional hypotheses. This is *in-context exploration*, i.e., search while uncertain, and exploitation once the hidden state is revealed. Here, on-policy data collection is essential as the student is trained on its own behavior, which provides coverage over test-time histories, while a curriculum over the student rollout horizon improves sample efficiency and enables longer-context adaptation (Section 4).

We show both empirically (Section 5) and theoretically (Appendix H) that ASTEROID approximates Bayesian posterior sampling, a provably efficient exploration strategy (Osband et al., 2013). We note that this exploration is not Bayes-optimal, i.e., it trades off some exploration efficiency for tractability and strong regret guarantees. However, scaling posterior sampling has been a bottleneck: prior approaches maintain an explicit posterior (Asmuth et al., 2009) or assume infinite offline data (Lee et al., 2023), which is intractable in realistic settings. ASTEROID generates this exploration behavior with neither drawback: it does not maintain an explicit posterior, and on-policy learning eliminates impractical data costs. In summary, we make the following contributions:

1. We present ASTEROID, a framework that pairs on-policy student contexts with privileged expert trajectories to learn in-context exploration and test-time improvement through supervised learning.
2. We evaluate our method across 11 environments, including navigation in realistic household simulators and real-world robot manipulation, where it outperforms RL and supervised baselines under fixed exploration budgets while using two orders of magnitude fewer training interactions.

3. We show that learned policies exhibit efficient real-world exploration and, under appropriate assumptions, recover posterior sampling with error linear rather than quadratic in context length.

2 Problem Statement

We study decision-making under *partial observability*, modelled as a partially observable Markov decision process (POMDP) (Kaelbling et al., 1998). A POMDP is represented by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, E, T, R, \rho_0, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{O}$ is the observation space and E, T, R, ρ_0, γ are the emission function, transition kernel, reward function, initial state distribution and discount factor, respectively. In a POMDP, the policy does not observe the state s_t but receives partial observations $o_t \sim E(\cdot \mid s_t)$. For simplicity, we also define the Bayes-Adaptive Markov decision process (BAMDP), a special case of POMDPs in which the only hidden variable is a latent parameter, denoted by z, i.e., $s_t = (o_t, z)$. For each interaction, $z \in \mathcal{Z}$ is sampled from a prior $p(z)$ and remains fixed for K episodes (unlike a POMDP, where hidden states may vary at each step). Given z, BAMDPs are fully observed and can be represented as a collection of MDPs M_z (or tasks).

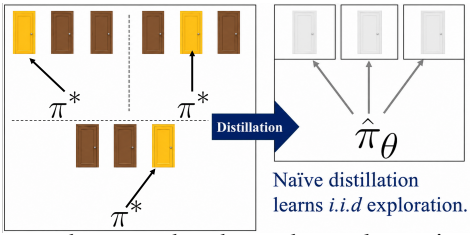
As a running example, consider a three-door environment: a BAMDP with latent parameter z denoting the target door. At the start of each interaction, z is sampled uniformly from the three doors and remains fixed across multiple episodes, where the objective is to navigate to the correct door. The target is hidden, and the agent must use its interaction history to infer the hidden goal and then exploit that information. We can construct a POMDP by adding noise to agent observations at each time step.

Since observations o_t are partial, optimal policies depend on the history $h_t = (o_0, a_0, r_0, \ldots, o_t)$. Let $d_\pi(h)$ denote the distribution over histories for a policy π. Learning information-gathering behavior directly from sparse rewards is highly sample-inefficient, often requiring impractical amounts of experience (Bauer et al., 2023). To mitigate this, we study an *asymmetric* training–testing setting in which a privileged teacher $\pi^*(a_t \mid s_t)$ can access s_t during training, while the deployed student $\pi_\theta(a_t \mid h_t)$ acts under partial observability using h_t. During training, the learner interacts with the POMDP to collect transition tuples (o_t, a_t, r_t, o_{t+1}) and queries the teacher for actions $a_t^* \sim \pi^*(\cdot \mid s_t)$. Our objective is to use these interactions and teacher supervision to learn a student $\pi_\theta(\cdot \mid h_t)$ that maximises the expected return $J(\pi_\theta) = \mathbb{E}_{\pi_\theta}[\sum_{t=0}^{\infty} \gamma^t r_t]$. [2] This asymmetric distillation setting arises naturally in navigation (Jain et al., 2021), sim-to-real robot manipulation (Yin et al., 2026), human-in-the-loop learning (Kelly et al., 2019), and knowledge distillation (Gu et al., 2024).

3 Background: Student-Teacher Distillation

Our work builds on student–teacher distillation (Swamy et al., 2022; Walsman et al., 2023; Bhardwaj et al., 2017), meta-learning (Duan et al., 2016; Zintgraf et al., 2021; Somani et al., 2013), and in-context reinforcement learning (Lee et al., 2023; Laskin et al., 2022; Chen et al., 2021). We defer a full related-work discussion to Appendix A. In the following sections, we cover the key background for teacher-student distillation under partial observability, explain why it breaks down in asymmetric settings, and motivate ASTEROID as a method for learning in-context exploration.

Naive asymmetric student–teacher distillation cannot learn adaptive exploration. A natural approach to train partially observed student policies is via behavior-cloning trajectories generated solely by a privileged policy Peng et al. (2018). But there is a conflict: the privileged teacher policy $\pi^*(\cdot|s_t)$ solves the task without exploration, whereas the unprivileged student policy π_θ must gather information through exploration before exploiting for task success (Hausknecht and Stone, 2015). Following the three-door example from Section 2, privileged experts observe z, but the student only receives partial observations. At $t = 0$, o_0 is identical for all values of z. Here, a student policy acting on



[2] Since π^* conditions on strictly more information, its performance upper-bounds that of any partially observed student.

partial observation o_0 cannot distinguish between actions provided for the three goals and learns the marginal expert behavior induced by the training data d_{π^*}, i.e., choosing a door uniformly at random.

Naive asymmetric distillation from expert trajectories learns *i.i.d.* exploration, but fails to exploit environment rewards and feedback.

In-context learning via Decision Pretrained Transformer (DPT) (Lee et al., 2023) can learn exploration. Lee et al. (2023) showed that asymmetric student-teacher distillation can learn exploration when the training dataset is constructed with specific design choices. It introduces DPT, an in-context learning framework that constructs a dataset of contexts, observations and target actions: $\mathcal{D}^{\text{DPT}} = \{(h_t, o_t, a_t^*)^i\}_{i=1}^N$ for supervised learning via random sampling. Here, $d_{\mathcal{D}}$ denotes the empirical distribution of histories and query observations in the dataset $\mathcal{D}$. DPT generates its offline histories with an independent behavior policy. Further, it labels randomly sampled query observations $o_t \in \mathcal{O}$ with the privileged expert $a_t^* \sim \pi^*(\cdot|o_t, z)$. The history h_t provides context about the underlying hidden parameter, which the policy learns to use at the query observation o_t to predict actions. DPT trains an in-context policy maximising action likelihood over this dataset :

$$\mathcal{L}(\theta) = \mathop{\mathbb{E}}_{(h_t, o_t, a_t^*) \sim \mathcal{D}} \left[-\log \pi_\theta(a_t^* \mid o_t, h_t) \right]. \tag{1}$$

Under coverage assumptions, DPT provably converges to posterior sampling: it uses h_t to form an implicit posterior over the latent state, and acts according to the expert policy averaged under this posterior, i.e., $\pi(a_t|h_t) = \int \pi^*(a_t|o_t, z)\ \mathrm{p}(z|h_t)\ dz$.

Let $\mathcal{H}_t = (\mathcal{O} \times \mathcal{A} \times \mathcal{R})^t \times \mathcal{O}$ denote the space of length-t history and observation tuples (h_t, o_t).

The primary limitation of DPT is the reliance on a full-coverage assumption: the dataset must cover the combinatorial space of arbitrary contexts and query observations, i.e., we require

$$\bigcup_{t=0}^{H} \mathcal{H}_t \subseteq \text{supp}(d_{\mathcal{D}}),$$

which scales exponentially with the interaction horizon H.

The above assumption holds in the infinite-data limit, when randomly sampling histories and query observations provides complete coverage (see Figure 2). Under finite data, this quickly becomes impractical in high-dimensional observation and action spaces, where the number of possible histories grows exponentially, and informative feedback is sparse (Bellman, 1966). Consequently, purely offline context datasets do not scale to learning exploration in realistic, large-scale decision-making problems. This has fundamentally limited the applicability of methods like DPT in robotics and other high-dimensional problems.

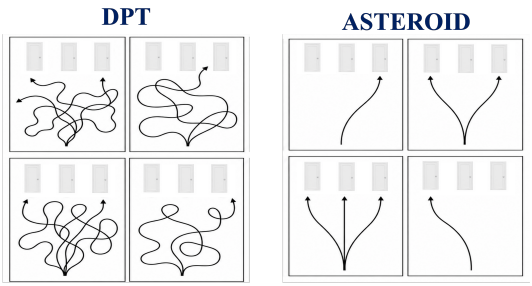


Figure 2: DPT intractably needs full-context coverage for training, while ASTEROID generates on-policy rollouts to learn scalable exploration.

4 ASYMMETRIC STUDENT TEACHER ROLL-OUT IN-CONTEXT DISTILLATION

In Section 3, we identified a central bottleneck in supervised learning for in-context exploration: generating data that covers the histories a partially observed agent will experience at deployment.

Our key insight is that the training dataset $\mathcal{D}$ should not necessarily cover the entire combinatorial history space $\mathcal{H}_t$; the distribution of trajectories generated by the student itself: the "on-policy" rollouts $h_t \sim d_{\pi_\theta}$ coupled with expert labels provides sufficient coverage in the dataset to learn an adaptive exploration strategy, i.e.

$$\text{supp}(d_{\pi_\theta}) \subseteq \text{supp}(d_{\mathcal{D}})$$

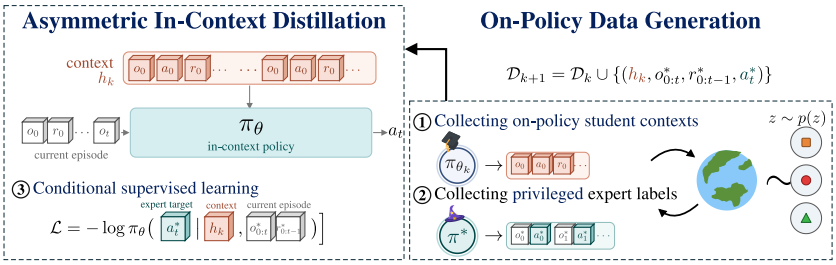


Figure 3: ASTEROID (1) collects on-policy context from the student π_{θ_k} and (2) expert trajectories from π^*, under the same hidden z. With this dataset, we (3) train policies that perform in-context exploration.

Consequently, this relaxes the strong assumption of infinite coverage over histories that renders DPT intractable. Our method ASTEROID is an iterative asymmetric distillation procedure that alternates between collecting on-policy histories from the student and training a history-conditioned student policy from privileged expert labels. Collecting supervision on student-induced histories reduces the distribution shift between training and deployment. Unlike standard interactive imitation (Ross et al., 2011), which aggregates per-step action labels, ASTEROID supervises with full expert trajectories and conditions on environment feedback (Appendix D).

4.1 How does ASTEROID learn in-context exploration?

Figure 3 and Algorithm 1 summarize ASTEROID. Let π_{θ_k} denote the student policy and $\mathcal{D}_k$ the dataset at iteration k. We first instantiate the method in a BAMDP, where a latent $z \sim p(z)$ stays fixed while the policy interacts for k episodes, and extend it to general POMDPs in Section 4.3. Each iteration has two phases. In the *data generation* phase, we roll out the student for k episodes to produce the on-policy context $h = (\tau^0, \ldots, \tau^{k-1})$, then roll out the privileged expert $\pi^*(\cdot \mid o, z)$ under the same z to produce an expert trajectory τ^*. The expert observations $o^*_{0:t}$ and rewards $r^*_{0:t-1}$ from τ^* serve as the query, and the expert action a^*_t as the label. Unlike DPT, which samples query observations randomly, our queries come from expert rollouts. In the *supervised learning* phase, we train π_{θ_k} to predict a^*_t given $(h, o^*_{0:t}, r^*_{0:t-1})$ with the DPT loss (Eq. 1). In practice, we use a decision transformer (Chen et al., 2021) and compute the loss only on expert actions. After iteration k, π_{θ_k} adapts using up to k test-time episodes. Revisiting the three-door problem from Section 2,

Iteration 0. π_{θ_0} learns an *i.i.d.* policy with no adaptation. It randomly picks a door every time, ignoring feedback.

Iteration 1. π_{θ_0} generates an on-policy trajectory visiting a random door. The reward is $+1$ if correct, 0 otherwise. The expert label then returns to the same door on success or points to a different door on failure. Thus, π_{θ_1} learns to use the first episode feedback as context for choosing the second. However, π_{θ_1} has a context window $= 1$, so it motivates further iterations for longer context adaptation.

Iteration 2. π_{θ_1} now generates two trajectories: it repeats a successful door, or tests a new door after failure. If both attempts fail, the expert label points to the remaining door. Thus, it learns in-context exploration that converges to the true goal. (see Appendix E)

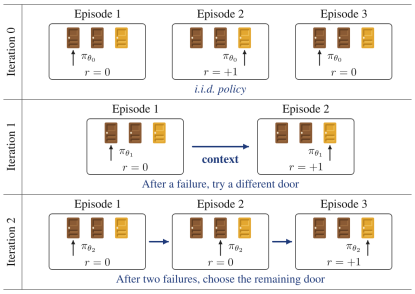


Figure 4: We iteratively learn in-context exploration over longer contexts.

4.2 Practical Implementation Details

Initialization. We initialize $\mathcal{D}_0$ with privileged expert trajectories. The initial policy π_{θ_0} is therefore trained without contextual information and resembles naive student–teacher distillation; see Section 3.

Deployment. At deployment, the policy starts from a null context, $h_0 = \emptyset$ (see Algorithm 3). We reuse no training context; at each step, the policy uses online interaction to build context and infer hidden information. For evaluations longer than the training context length, we use a sliding window.

Context-length curriculum. At each iteration k, the policy is rolled out for k episodes, inducing a curriculum on rollout horizons. As the horizon expands, the student can eliminate task instances that

are inconsistent with its history over a longer horizon, making exploration efficient. This also makes our method more sample efficient, and we discuss this further in Appendix B.3.

4.3 How does ASTEROID work on general POMDPs?

We described ASTEROID in a BAMDP, where a latent variable z remains fixed across multiple episodes. In a general POMDP, the hidden state evolves within a single episode. For example, in a locally observed maze, the agent does not observe its global position, and this hidden position changes after every move. At test time, the policy interacts with the environment for a single episode over a fixed horizon H. As the underlying state changes at each step, the agent faces a branching inference problem: each action and observation changes which hidden states remain plausible. Thus, instead of coverage over repeated episodes for a fixed z, we need coverage over on-policy histories h_t across timesteps $t \in [H]$. In practice (see Algorithm 2), ASTEROID samples a timestep t, rolls out the student to produce an on-policy history h_t, and queries the expert from the reached state, producing a partial trajectory that generates the queries $o^*_{t:}$ and action labels $a^*_{t:}$. The resulting $(h_t, o^*_{t:}, a^*_{t:})$ train the policy to map its history to actions that are optimal for the hidden state under that history. At the takeover step, it approximates $\pi(a_t \mid h_t) = \int \pi^*(a_t \mid s_t)\, p(s_t \mid h_t)\, ds_t$, and over the future steps, the expert trajectory teaches it to follow one hypothesis rather than dithering. For example, in the maze, after each move, the local observations and rewards prune some possible positions while keeping others plausible. Further, we induce a curriculum in the context horizon by iteratively biasing the sampled timestep t toward longer horizons. Thus, we adapt the method to a POMDP: collect on-policy histories, then use privileged labels to teach it how to guide future behaviour.

5 Experiments

In our experiments, we investigate: 1) Does ASTEROID learn to solve partially observed problems efficiently on deployment? 2) Is it sample-efficient to train as compared to baselines? 3) What exploration behavior do ASTEROID policies display at test time? 4) Does it learn exploration policies that can be deployed on real-robot hardware? 5) Is learning from on-policy contexts necessary?

5.1 Training and Evaluation Details

We evaluate all methods across 11 environments and three seeds [3]. For model architecture and tokenisation, we use the causal transformer implementation from Lee et al. (2023). We include the hyperparameters, training, and environment details in Appendix F. We also include ablations over ASTEROID in Appendix B, and qualitative videos on the website[4]. We evaluate the policy's performance and exploration efficiency across multiple test-time episodes on held-out goals. We compare policy performance across training iterations, with a fixed test-time budget.

Tasks. We evaluate our approach on a diverse suite of partially observable decision-making problems across robotic manipulation, discrete exploration, and continuous navigation (Fig. 5), requiring strong elements of exploration, memory, and adaptation under limited sensing. **Partially observed gridworlds Laskin et al. (2022)** are search problems where an agent acts for multiple episodes in a gridworld to reach one or more goals. **Meta-Learning navigation**: an agent must navigate in an environment with hidden goals. We instantiate (i) Ant Navigation Rakelly et al. (2019) and (ii) Procgen Maze Navigation (Cobbe et al., 2019). **Simulated Robotic Manipulation Yin et al. (2026)**: continuous-control manipulation under limited sensing (i.e. wrist camera or tactile): Wrist-Peg-Insertion and Tactile-Cube-Pick. **Embodied Object Search Puig et al. (2024)** involves searching for household objects in Habitat-Sim and presents a realistic, visually complex domain. **Real Robot Manipulation** We construct a real-robot task: pick up a cube using only proprioception from the arm and gripper. We train policies in simulation and then transfer them to the real world.

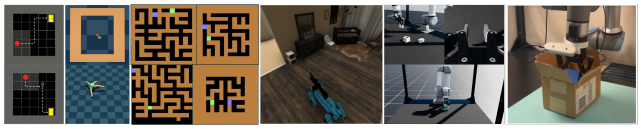


Figure 5: Tasks: (left to right) meta-learning gridworld, partially observed navigation, Procgen maze navigation, embodied object search, simulated robot and real robot manipulation.

[3] We run Habitat on a single seed due to high wall-clock time.

[4] Project webpage: sites.google.com/view/icl-exploration

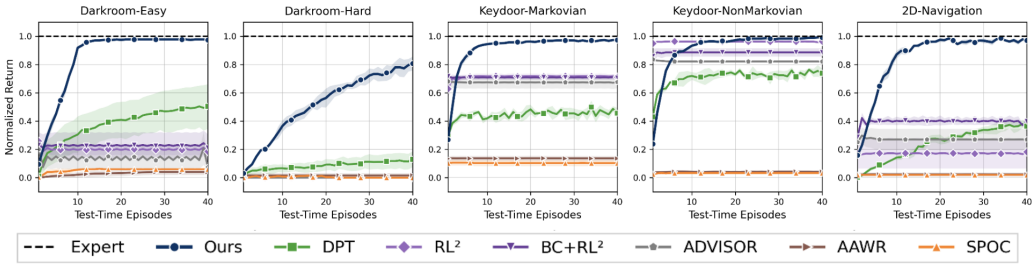


Figure 6: Performance on discrete exploration tasks requiring memory and goal inference. We observe that ASTEROID outperforms all baselines in terms of test-time efficiency. It performs efficient exploration and almost converges to the expert after inferring the hidden task variables.

Baselines. We compare against ten baselines from several frameworks under partial observability: (i) **Asymmetric student–teacher: BC**: imitates privileged expert trajectories, **DAgger**: relabels the student's per-step actions with the expert. **DPT** Lee et al. (2023) trains a transformer on an in-context dataset of randomly collected interactions to predict oracle actions; **SPOC** Ehsani et al. (2023) imitates shortest-path trajectories with data augmentation, yielding exploration-like behavior. **AAWR** Hu et al. (2025) extends Advantage Weighted Regression Peng et al. (2019) to POMDPs, using privileged state during training to learn a stronger critic. (ii) **Meta-RL: RL**^{2} (Duan et al., 2016) trains a recurrent policy to adapt to the hidden parameters purely through interaction, treating the policy's hidden state as an implicit learning algorithm that updates from histories across episodes. **Distilled Planner** Puig et al. (2024); Osband et al. (2013): For embodied navigation in Habitat, we train a policy that approximates discrete posterior-sampling RL (PSRL) Osband et al. (2013) (i.e., upper bound performance). (iii) **Imitation-informed RL: BC+PPO** Schulman et al. (2017); Duan et al. (2016) accelerates PPO Schulman et al. (2017) by adding a behavior-cloning loss over privileged actions; for meta-learning tasks, we use an adapted variant (**BC+RL**^{2}). **ADVISOR** Weihs et al. (2021) dynamically combines IL and online RL objectives during training, resolving the imitation gap adaptively. **RMA** Kumar et al. (2021) learns to predict the privileged latent from history, and a frozen expert policy acts on that predicted latent.

5.2 Does ASTEROID solve partially observed problems efficiently?

In Figure 6, we compare the test-time performance of in-context policies learned via ASTEROID against baselines. Given the same test-time budget, our method achieves substantially higher final performance. Unlike the baselines, it covers relevant test-time histories and can explore effectively to infer task-relevant information. Figures 8, 12, and 13 show the same trend in partially observed navigation and manipulation. In Procgen Nav (Figure 7), ASTEROID solves procedurally generated mazes by using local observation history to maintain an implicit belief over the state, guiding exploration toward unexplored regions (Karkus et al., 2017). This avoids redundant search and enables ASTEROID to achieve higher success while being $2\times$ faster. In contrast, BC achieves non-trivial success via random exploration but is never trained on informative contexts and is highly inefficient. DPT underperforms because offline datasets provide incomplete coverage over test-time histories. RMA does passive exploration by predicting the privileged latent from history, and a frozen base policy acts on that prediction. Under partial observability, latent uncertainty generates OOD inputs for the base policy, which is trained only under full observability. ASTEROID instead directly supervises the action distribution conditioned on the student's own context to perform directed exploration. In ASTEROID, exploration is the learned behaviour, not a byproduct. Finally, as we discuss in Appendix D, DAgger under partial observability degrades performance (Figure 8), while history conditioning and multi-step supervision in ASTEROID learns effective search. We show robustness to imperfect experts in Appendix B.2.

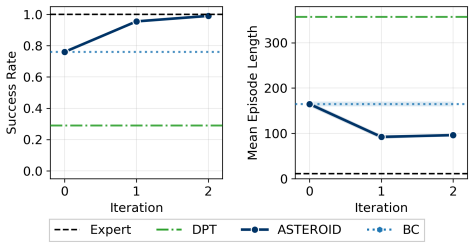


Figure 7: ASTEROID improves success while being efficient on Procgen; BC succeeds non-trivially via random exploration but is $\sim 2\times$ slower.

5.3 Is ASTEROID sample efficient and simple to train?

In this section, we compare the training-time sample efficiency of ASTEROID against baselines that combine RL with privileged supervision to learn policies under partial observability. Using GPU-parallelized simulation Mittal et al. (2023), we evaluate manipulation tasks in which the policy has access to limited sensory modalities. In Figure 8, ASTEROID is $\sim 100\times$ more efficient in environment steps than RL baselines, learning a policy that explores and successfully picks the cube or inserts the peg. In Figure 13b, the student observes only the history of past actions and tactile sensing (no proprioception), and ASTEROID efficiently learns policies with $\sim 92\%$ success rate under this partial observability. The supervised updates in ASTEROID (as compared to expensive RL) make it highly efficient during training, even for high-dimensional tasks. While RL baselines can learn in low-dimensional settings (Appendix B.4), they are significantly sample-inefficient in domains such as navigation and robotics (Figures 8 and 12).

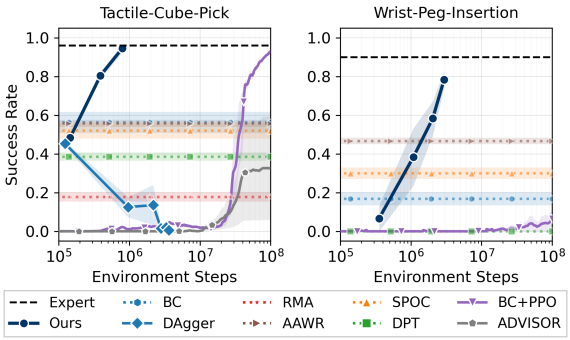


Figure 8: We observe extreme sample efficiency $\sim 100\times$ speedup when using ASTEROID to train policies for robot manipulation under limited sensing.

5.4 What kind of in-context exploratory behavior do policies display?

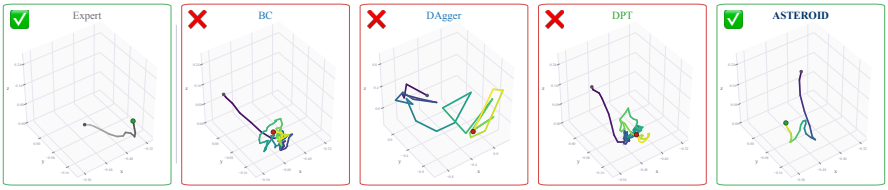
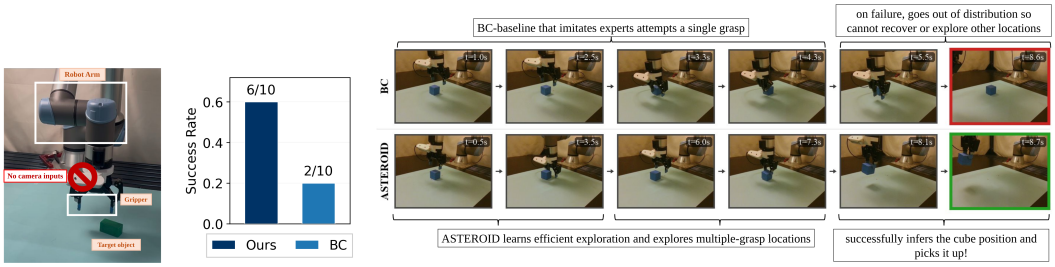


Figure 9: End-effector trajectories on tactile manipulation: the policy must locate and lift a cube using touch. The expert (gray) observes the cube pose and grasps it directly. Under partial observability, baselines fail to adapt. ASTEROID explores the table until contact, then grasps and lifts the cube.

In Tactile-Pick (Figure 9), the policy probes multiple grasp points, uses tactile feedback from failed attempts to inform its next actions; unlike BC, which treats trials independently and relies on i.i.d. exploration. Additionally, other baselines fail to learn directed exploration either because of incompatible supervision or inefficient context generation during training. Similarly, in Figure 18, the robot policy scans the tabletop using its wrist-camera to localize both the object and target. In contrast, naive distillation goes out of distribution and degenerates. We also see goal-directed, memory-based exploration (Appendix C.2) in Habitat and navigation tasks: the policy explores the environment, remembers failed locations, and avoids redundant search. See Appendix C and the website for videos. In Figure 12, we observe in the Ant Navigation task that the learned exploration resembles strategies found in prior meta-learning methods (Zintgraf et al., 2021). Although RL can learn exploration, its sample inefficiency makes it impractical. DPT performs poorly in continuous control, as it is nearly impossible to construct offline datasets via random exploration that cover test-time histories.

5.5 Do ASTEROID exploration policies transfer to real robot hardware?

In Figure 10a, we evaluate ASTEROID on a real robot manipulation task that requires exploration under partial observability. We first train clairvoyant experts in IsaacLab using large-scale reinforcement learning with domain randomization (Mittal et al., 2023; Zhang et al., 2026). These experts use privileged state information to pick up a cube. At deployment, the real robot observes only proprioception (joint angles and gripper positions, no touch). The cameras are off, so it cannot see the cube, and we cannot deploy vision-based policies. We train student policies to explore the world and infer this hidden information by trying different grasp locations and using gripper jaw position. The standard transfer baseline is to distill the simulation expert into the student policy via behavior cloning. This student policy imitates expert trajectories, but does not learn how to recover from



(a) (b) (c)

Figure 10: (a) Real robot setup, where it must rely solely on proprioception (arm joint and gripper positions) to infer the object's location and pick it up. (b) Comparing sim-to-real via BC and our method. We learn effective exploration that achieves higher success. (c) Behavior learned with BC vs ASTEROID. The baseline (first row) is trained to imitate expert data only and goes out of distribution. Meanwhile, ASTEROID (second row) learns a policy that uses its in-context exploration to explore multiple grasp locations and pick up the object.

failed attempts. In contrast, ASTEROID trains the policy to perform in-context exploration, so the robot learns to adapt its next grasp attempt based on the history of previous attempts. As shown in Figure 10b, ASTEROID achieves a higher success rate than behavior cloning. The qualitative rollouts in Figure 10c show this behavior: after a failed attempt, the baseline goes out of distribution, while ASTEROID continues searching over potential cube locations using its in-context abilities, completes the grasp, and successfully picks up the cube without cameras. We further show that as the policy does not use vision, it can successfully solve the task in situations where the partial observability is natural, e.g., rummaging inside a box or under visual distractors (see robot videos on our website).

5.6 Is learning from on-policy contexts necessary?

Across multiple domains, DPT performs poorly because offline random contexts fail to capture the deployment histories. In Figure 11, we compare ASTEROID against AAWR Hu et al. (2025) trained with offline datasets of varying diversity (simulated via noise) in the peg-insertion task, to highlight the importance of on-policy contexts. AAWR, which learns from an offline dataset, is highly sensitive to the dataset quality. ASTEROID avoids this offline dependence by generating on-policy contexts from the student.

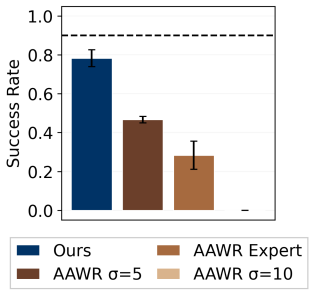


Figure 11: ASTEROID is online, while AAWR is sensitive to quality of offline datasets.

6 Discussion

In this work, we propose ASTEROID, an iterative *asymmetric* distillation framework for learning in-context exploration. The key idea is to train a history-conditioned policy using supervised targets from a privileged expert, while collecting the histories *on-policy* from the student. This directly addresses two bottlenecks for exploration: (i) privileged teachers do not demonstrate information-gathering behaviour, and (ii) offline or randomly collected histories may not match the distribution the student will encounter at deployment. While the need for history-conditioned exploration policies is well established, we show that on-policy distillation from privileged teachers, without RL or latent regression, is sufficient to learn exploration scalably across diverse simulated and real-world tasks.

A limitation is that the approach relies on a strong privileged expert, though it is agnostic to how the expert is generated. An interesting direction is to remove this reliance and train experts from offline demonstrations. While posterior sampling has sublinear regret, it can be inefficient, motivating future extensions that combine our approach with optimistic exploration. Another limitation is that posterior-sampling exploration relies on inferring hidden states from the overall expert support. When the hidden information can only be gathered outside this support, the student cannot reduce its uncertainty. A promising future direction is to learn a residual policy on top of this student to learn off-support information-gathering behavior. Additionally, the student policy is trained to imitate expert actions, but may produce unsafe exploration. Therefore, an important direction is to enforce safety via explicit constraints and safety filters.

7 AI USE

In this work, we used generative AI tools to help implement and debug the experiment codebase, including porting prior baselines to IsaacLab frameworks, processing results, and producing figures. We also used AI tools to edit portions of the manuscript, verify the proof sketches, and find potentially relevant literature. We did not use generative AI tools to develop the conceptual framework; the authors formulated the original problem statement, sourced the benchmarks, performed the analysis, conducted the experiments, and ran the real-robot experiments. Generative AI was not used to format datasets, generate synthetic data, or interpret results. We manually verified all suggested references against their original sources. We reviewed and tested AI-assisted code, checked figures against the underlying seed-level results, and manually revised the manuscript. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

8 ETHICS AND REPRODUCIBILITY STATEMENT

We introduce algorithms for improving in-context exploration and test them in simulated domains and a controlled real-world experiment. This work uses only simulated environments and a real-robot manipulation task, and does not involve human subjects or personal data. Since the learned policies explore by trial and error, real-world deployment may lead to unsafe exploration, and we see enforcing safety via explicit constraints and safety filters as an important direction (Section 6). In Section 5 and the Appendix, we carefully outline all relevant details, including task setups, parameters, architecture details, and training details to enable reproducibility. We run almost all experiments across multiple seeds, and will release complete logs and code for the method and all baselines in the camera-ready version.

REFERENCES

A. Agarwal, S. Kakade, A. Krishnamurthy, and W. Sun. Flambe: Structural complexity and representation learning of low rank mdps, 2020. URL https://arxiv.org/abs/2006.10814.

J. Asmuth, L. Li, M. L. Littman, A. Nouri, and D. Wingate. A bayesian sampling approach to exploration in reinforcement learning. UAI '09, page 19–26, Arlington, Virginia, USA, 2009. AUAI Press. ISBN 9780974903958.

M. Bansal, A. Krizhevsky, and A. Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018.

J. Bauer, K. Baumli, F. Behbahani, A. Bhoopchand, N. Bradley-Schmieg, M. Chang, N. Clay, A. Collister, V. Dasagi, L. Gonzalez, K. Gregor, E. Hughes, S. Kashem, M. Loks-Thompson, H. Openshaw, J. Parker-Holder, S. Pathak, N. Perez-Nieves, N. Rakicevic, T. Rocktäschel, Y. Schroecker, S. Singh, J. Sygnowski, K. Tuyls, S. York, A. Zacherl, and L. Zhang. Human-timescale adaptation in an open-ended task space. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

R. Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966. doi: 10.1126/science.153.3731.34.

M. Bhardwaj, S. Choudhury, and S. Scherer. Learning heuristic search via imitation. In *Conference on Robot Learning*, pages 271–280. PMLR, 2017.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

O. Chapelle and L. Li. An empirical evaluation of thompson sampling. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/file/e53a0a2978c28872a4505bdb51db06dc-Paper.pdf.

D. S. Chaplot, D. Gandhi, A. Gupta, and R. Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *In Neural Information Processing Systems (NeurIPS)*, 2020a.

D. S. Chaplot, R. Salakhutdinov, A. K. Gupta, and S. Gupta. Neural topological slam for visual navigation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12872–12881, 2020b. URL https://api.semanticscholar.org/CorpusID:214754592.

D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl. Learning by cheating. In *Conference on robot learning*, pages 66–75. PMLR, 2020.

L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *arXiv preprint arXiv:2106.01345*, 2021.

M. Chevalier-Boisvert, B. Dai, M. Towers, R. Perez-Vicente, L. Willems, S. Lahlou, S. Pal, P. S. Castro, and J. Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. In *Advances in Neural Information Processing Systems 36, New Orleans, LA, USA*, December 2023.

S. Choudhury, A. Kapoor, G. Ranade, S. Scherer, and D. Dey. Adaptive information gathering via imitation learning. *arXiv preprint arXiv:1705.07834*, 2017.

K. Cobbe, C. Hesse, J. Hilton, and J. Schulman. Leveraging procedural generation to benchmark reinforcement learning. *arXiv preprint arXiv:1912.01588*, 2019.

H. Daumé Iii, J. Langford, and D. Marcu. Search-based structured prediction. *Machine learning*, 75 (3):297–325, 2009.

P. De Haan, D. Jayaraman, and S. Levine. Causal confusion in imitation learning. *Advances in neural information processing systems*, 32, 2019.

Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel. Rl^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

K. Ehsani, T. Gupta, R. Hendrix, J. Salvador, L. Weihs, K.-H. Zeng, K. P. Singh, Y. Kim, W. Han, A. Herrasti, R. Krishna, D. Schwenk, E. VanderBilt, and A. Kembhavi. Imitating shortest paths in simulation enables effective navigation and manipulation in the real world. *arXiv*, 2023.

C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1126–1135. JMLR.org, 2017.

L. Fu, H. Huang, G. Datta, L. Y. Chen, W. C.-H. Panitch, F. Liu, H. Li, and K. Goldberg. In-context imitation learning via next-token prediction. *arXiv preprint arXiv:2408.15980*, 2024.

S. Garg, D. Tsipras, P. S. Liang, and G. Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in neural information processing systems*, 35:30583–30598, 2022.

Y. Gu, L. Dong, F. Wei, and M. Huang. Minillm: Knowledge distillation of large language models. In *The twelfth international conference on learning representations*, 2024.

M. J. Hausknecht and P. Stone. Deep recurrent q-learning for partially observable mdps. In *AAAI fall symposia*, volume 45, page 141, 2015.

C. Higuera, J. Ortiz, H. Qi, L. Pineda, B. Boots, and M. Mukadam. Perceiving extrinsic contacts from touch improves learning insertion policies. *arXiv preprint arXiv:2309.16652*, 2023.

E. S. Hu, J. Wang, X. Yuan, F. Luo, M. Li, G. Lambrechts, O. Rybkin, and D. Jayaraman. Real-world reinforcement learning of active perception behaviors. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=RkdTtznSAL.

M. Igl, L. M. Zintgraf, T. A. Le, F. Wood, and S. Whiteson. Deep variational reinforcement learning for pomdps. In *International Conference on Machine Learning*, 2018. URL https://api.semanticscholar.org/CorpusID:46955236.

P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

U. Jain, I.-J. Liu, S. Lazebnik, A. Kembhavi, L. Weihs, and A. G. Schwing. Gridtopix: Training embodied agents with minimal supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15141–15151, 2021.

L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134, 1998. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(98)00023-X. URL https://www.sciencedirect.com/science/article/pii/S000437029800023X.

P. Karkus, D. Hsu, and W. S. Lee. Qmdp-net: Deep learning for planning under partial observability. *Advances in neural information processing systems*, 30, 2017.

M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8077–8083. IEEE, 2019.

Y. Kim, N. Chin, A. Vasudev, and S. Choudhury. Distilling realizable students from unrealizable teachers. *arXiv preprint arXiv:2505.09546*, 2025.

A. Kumar, Z. Fu, D. Pathak, and J. Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

M. Laskin, L. Wang, J. Oh, E. Parisotto, S. Spencer, R. Steigerwald, D. Strouse, S. Hansen, A. Filos, E. Brooks, M. Gazeau, H. Sahni, S. Singh, and V. Mnih. In-context reinforcement learning with algorithm distillation, 2022. URL https://arxiv.org/abs/2210.14215.

G. Lee, B. Hou, S. Choudhury, and S. S. Srinivasa. Bayesian residual policy optimization:: Scalable bayesian reinforcement learning with clairvoyant experts. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5611–5618. IEEE, 2021.

J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

J. N. Lee, A. Xie, A. Pacchiano, Y. Chandak, C. Finn, O. Nachum, and E. Brunskill. Supervised pretraining can learn in-context reinforcement learning. *arXiv preprint arXiv:2306.14892*, 2023.

E. Z. Liu, A. Raghunathan, P. Liang, and C. Finn. Decoupling exploration and exploitation for meta-reinforcement learning without sacrifices. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6925–6935. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/liu21s.html.

H. Liu and P. Abbeel. Emergent agentic transformer from chain of hindsight experience. In *International Conference on Machine Learning*, pages 21362–21374. PMLR, 2023.

S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 11048–11064, 2022.

M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, et al. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023.

H. Nguyen, A. Baisero, D. Wang, C. Amato, and R. Platt. Leveraging fully observable policies for learning under partial observability. *arXiv preprint arXiv:2211.01991*, 2022.

T. Ni, B. Eysenbach, and R. Salakhutdinov. Recurrent model-free rl can be a strong baseline for many pomdps. *arXiv preprint arXiv:2110.05038*, 2021.

A. Nikulin, I. Zisman, A. Zemtsov, and V. Kurenkov. Xland-100b: A large-scale multi-task dataset for in-context reinforcement learning, 2025. URL https://arxiv.org/abs/2406.08973.

I. Osband, D. Russo, and B. Van Roy. (more) efficient reinforcement learning via posterior sampling. *Advances in Neural Information Processing Systems*, 26, 2013.

X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018.

X. B. Peng, A. Kumar, G. Zhang, and S. Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *CoRR*, abs/1910.00177, 2019. URL https://arxiv.org/abs/1910.00177.

X. Puig, E. Undersander, A. Szot, M. D. Cote, T.-Y. Yang, R. Partsey, R. Desai, A. Clegg, M. Hlavac, S. Y. Min, V. Vondrus, T. Gervet, V.-P. Berges, J. M. Turner, O. Maksymets, Z. Kira, M. Kalakrishnan, J. Malik, D. S. Chaplot, U. Jain, D. Batra, A. Rai, and R. Mottaghi. Habitat 3.0: A co-habitat for humans, avatars, and robots. In *ICLR*. OpenReview.net, 2024. URL http://dblp.uni-trier.de/db/conf/iclr/iclr2024.html#PuigUSCYPDCHMVG24.

K. Rakelly, A. Zhou, C. Finn, S. Levine, and D. Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5331–5340. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/rakelly19a.html.

S. Ross and J. A. Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014.

S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://dblp.uni-trier.de/db/journals/corr/corr1707.html#SchulmanWDRK17.

A. Somani, N. Ye, D. Hsu, and W. S. Lee. Despot: Online pomdp planning with regularization. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/c2aee86157b4a40b78132f1e71a9e6f1-Paper.pdf.

J. Spencer, S. Choudhury, A. Venkatraman, B. Ziebart, and J. A. Bagnell. Feedback in imitation learning: The three regimes of covariate shift. *arXiv preprint arXiv:2102.02872*, 2021.

W. Sun, A. Venkatraman, G. J. Gordon, B. Boots, and J. A. Bagnell. Deeply AggreVaTeD: Differentiable imitation learning for sequential prediction. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3309–3318. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/sun17d.html.

G. Swamy, S. Choudhury, J. Bagnell, and S. Z. Wu. Sequence model imitation learning with unobserved contexts. *Advances in Neural Information Processing Systems*, 35:17665–17676, 2022.

G. Tennenholtz, A. Hallak, G. Dalal, S. Mannor, G. Chechik, and U. Shalit. On covariate shift of latent confounders in imitation and reinforcement learning. *arXiv preprint arXiv:2110.06539*, 2021.

W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25:285–294, 1933. doi: 10.1093/biomet/25.3-4.285. URL https://doi.org/10.1093/biomet/25.3-4.285.

L. von Werra, Y. Belkada, L. Tunstall, E. Beeching, T. Thrush, N. Lambert, S. Huang, K. Rasul, and Q. Gallouédec. TRL: Transformers Reinforcement Learning, 2020. URL https://github.com/huggingface/trl.

A. Walsman, M. Zhang, S. Choudhury, D. Fox, and A. Farhadi. Impossibly good experts and how to follow them. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sciA_xgYofB.

Y. Wan, J. Wu, M. Abdulhai, L. Shani, and N. Jaques. Enhancing personalized multi-turn dialogue with curiosity reward. *arXiv preprint arXiv:2504.03206*, 2025.

J. X. Wang, Z. Kurth-Nelson, D. Tirumala, H. Soyer, J. Z. Leibo, R. Munos, C. Blundell, D. Kumaran, and M. Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

A. Warrington, J. W. Lavington, A. Scibior, M. Schmidt, and F. Wood. Robust asymmetric learning in pomdps. In *International Conference on Machine Learning*, pages 11013–11023. PMLR, 2021.

L. Weihs, U. Jain, I.-J. Liu, J. Salvador, S. Lazebnik, A. Kembhavi, and A. Schwing. Bridging the imitation gap by adaptive insubordination. In *NeurIPS*, 2021. the first two authors contributed equally.

P. Yin, T. Westenbroek, Z. Zhang, J. Tran, I. Dagnino, E. Shilamkar, N. Mbiziwo-Tiapo, S. Bagaria, X. Liu, G. Mullins, A. Kolobov, and A. Gupta. Emergent dexterity via diverse resets and large-scale reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Y. You, V. Thomas, A. Schutz, R. Skilton, N. Hawes, and O. Buffet. Partially observable monte-carlo graph search. *Proceedings of the International Conference on Automated Planning and Scheduling*, 35(1):279–287, Sep. 2025. doi: 10.1609/icaps.v35i1.36129. URL https://ojs.aaai.org/index.php/ICAPS/article/view/36129.

T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

Z. Zhang, M. Guaman Castro, P. Yin, A. Gupta, R. Scalise, and B. Boots. A balanced data diet: Addressing the bottleneck in mega-scale rl for robot control, 2026. Conference on Robot Learning.

L. Zintgraf, S. Schulze, C. Lu, L. Feng, M. Igl, K. Shiarlis, Y. Gal, K. Hofmann, and S. Whiteson. Varibad: Variational bayes-adaptive deep rl via meta-learning. *Journal of Machine Learning Research*, 22(289):1–39, 2021. URL http://jmlr.org/papers/v22/21-0657.html.

APPENDIX CONTENTS

A Related Work **16**

B Ablation Experiments **17**

B.1 Performance of ASTEROID on additional environments. 17

B.2 Robustness of ASTEROID to noise in privileged expert supervision 17

B.3 How does the curriculum of exploration affect ASTEROID training? 18

B.4 ASTEROID is more scalable than RL in complex domains. 18

B.5 Does iterative context accumulation actually help ASTEROID? 19

B.6 How well does ASTEROID generalize under shifted latent priors? 19

C Qualitative Results **19**

C.1 ASTEROID learns efficient exploration in robot manipulation under limited sensing, and Navigation. 19

C.2 ASTEROID presents emergent in-context properties such as memory as a result of on-policy in-context learning. 20

D Per-step interactive imitation learning fails in history-conditioned asymmetric distillation. **21**

E Didactic Example: ASTEROID learns efficient posterior sampling. **21**

F Training and Implementation Details **22**

F.1 Training Details 22

F.2 Tasks 23

G Baseline Implementation Details **25**

G.1 Pseudocode 26

G.2 Hyperparameters 26

H Additional Theory **27**

H.1 Preliminaries 28

H.2 Privileged Labels Provide Posterior-Sampling Supervision 28

H.3 Offline versus On-Policy Training 29

A Related Work

Student–Teacher Distillation. Learning from privileged experts arises in settings from self-driving (Chen et al., 2020; De Haan et al., 2019; Bansal et al., 2018) to robotic exploration (Choudhury et al., 2017; Jain et al., 2021; Bhardwaj et al., 2017). In partially observed domains, interactive imitation learning addresses covariate shift and the resulting compounding errors that drive the learner into unseen states (Chen et al., 2020; Lee et al., 2020) with classic iterative schemes including (Daumé Iii et al., 2009; Ross et al., 2011; Ross and Bagnell, 2014; Sun et al., 2017). Recent results suggest that on-policy imitation is necessary under partial observability (Swamy et al., 2022), and that off-policy cloning (Liu and Abbeel, 2023) can fail in non-realisable settings where information-gathering behaviour is required (Spencer et al., 2021; Weihs et al., 2021). Our work targets this non-realisable regime: the learner must take information-gathering actions that are not directly recoverable from a fully observed expert (Lee et al., 2021). Prior approaches handle this via additional assumptions (e.g., access to confounders at test time (Tennenholtz et al., 2021) or differentiable experts (Warrington et al., 2021)) or by incorporating reinforcement learning to induce exploration (Nguyen et al., 2022; Weihs et al., 2021; Walsman et al., 2023; Hu et al., 2025; Weihs et al., 2021). RMA (Kumar et al., 2021) learns to predict the privileged latent from history, and a frozen base policy acts on that prediction. This works when the history is passively informative. But it fails when the task needs directed exploration. There, the history reveals nothing unless the policy actively gathers information. In ASTEROID, exploration is the learned behaviour, not a byproduct. In contrast, we learn information-gathering behaviour using iterative distillation conditioned on on-policy histories, which is substantially more stable and sample-efficient than RL-based alternatives.

Meta Learning. A common approach to learning in POMDPs is to use recurrent policies that compress the interaction history into a latent belief representation (Hausknecht and Stone, 2015; Igl et al., 2018). Meta-RL provides a general framework for learning adaptive exploration across task distributions (Duan et al., 2016; Zintgraf et al., 2021; Rakelly et al., 2019), but often requires extensive online training interaction (Wang et al., 2016; Bauer et al., 2023), making it difficult to scale to high-dimensional, sparse-reward domains with long-horizon credit assignment. Our approach replaces this expensive RL-based adaptation with iterative supervised learning, still enabling long-horizon, history-dependent exploration through on-policy context collection. Additionally, unlike prior work (Lee et al., 2023; Zintgraf et al., 2021; Duan et al., 2016; Lee et al., 2021), our method is not limited to Bayes-Adaptive MDPs (Lee et al., 2021), where the agent interacts with an environment with a fixed underlying parameter across multiple episodes. We show that our method scales to POMDPs, where the policy explores actions within a single episode to estimate the underlying state distribution and achieve task success.

Posterior Sampling. Posterior sampling methods (Thompson, 1933; Osband et al., 2013) drive exploration by implicitly or explicitly maintaining a belief over the environment's underlying hidden parameters, acting optimally under samples from this distribution, and updating the belief based on feedback. Despite strong empirical and theoretical results (Chapelle and Li, 2011; Osband et al., 2013), such approaches typically require maintaining and sampling from an explicit posterior, and solving for policies under the sampled parameters (e.g., PSRL, BOSS) (Osband et al., 2013; Asmuth et al., 2009), limiting practicality in rich-observation settings. While recent approaches Somani et al. (2013); You et al. (2025) learn Bayes-optimal policies for POMDPs in discrete settings, posterior sampling yields a simple, scalable solution with provable regret guarantees. ASTEROID does not explicitly optimise a posterior-sampling objective, but in Section 5 we show that posterior-sampling-like behaviour emerges naturally through iterative on-policy distillation, even in high-dimensional continuous environments. Under the stated assumptions, we show that ASTEROID approximates posterior sampling on the covered history support. We also observe this behaviour empirically in high-dimensional manipulation, and it transfers to a real robot (see Section 5).

In-Context Decision-Making Recent works (Intelligence et al., 2025; Brown et al., 2020; von Werra et al., 2020) extensively use transformers as decision-making policies. When trained on a large corpus of sequential data using autoregressive supervised prediction, transformers can exhibit in-context learning (Fu et al., 2024; Min et al., 2022; Garg et al., 2022), i.e., adapting their behavior to new inputs based on interaction histories or input-output examples provided in context. In online decision-making, prior works have used these emergent capabilities to steer policies conditioned on rewards or prior context. For example, Decision Transformer (DT) Chen et al. (2021) learns a transformer to autoregressively predict actions conditioned on past interactions and the desired

return. Similarly, Algorithm Distillation Laskin et al. (2022) uses experience traces from a learning method (such as PPO (Schulman et al., 2017)) to adapt and improve policy behavior based on past contexts. More recently, Decision Pretrained Transformer (Lee et al., 2023) shows that we can learn an in-context policy that performs posterior sampling in certain POMDPs. DPT generates the context using random interactions in the environment and action supervision from oracle experts with privileged information. Such methods assume these offline datasets cover all possible interaction histories, which is intractable and unrealistic in high-dimensional domains. ASTEROID instead uses a dataset of on-policy contexts, removing the coverage assumption and scaling to much harder, higher-dimensional problems (see Section 5).

B Ablation Experiments

B.1 Performance of ASTEROID on additional environments.

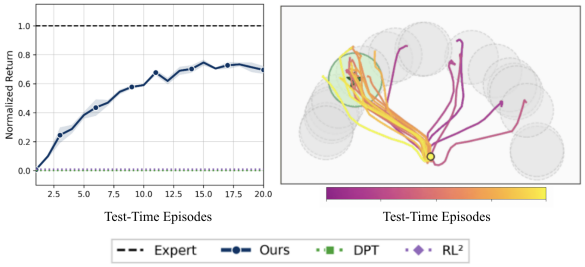
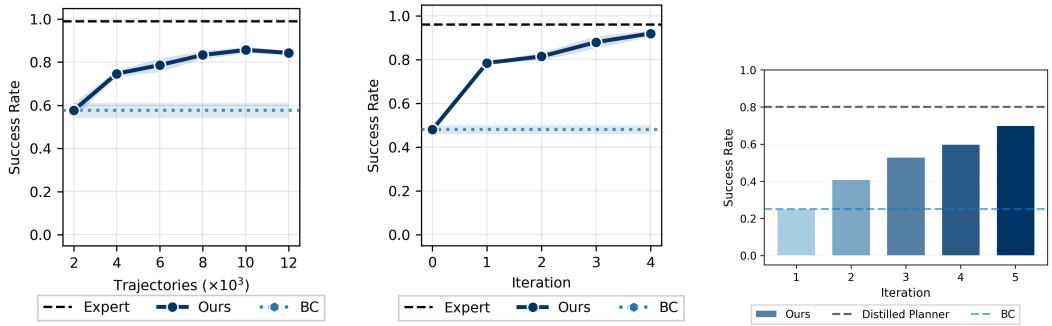


Figure 12: Ant Navigation. (Right) ASTEROID performs posterior sampling over goals while controlling the 8-dof robot and achieves high success (Left), while RL^2 and DPT (success rate ~ 0) fail to learn control and exploration for higher-dimensional action spaces.

In Figure 13, we show the performance of our proposed methods across additional partially observed environments. ASTEROID leverages on-policy context accumulation and expert supervision to solve these tasks efficiently. In the Tactile Pick w/o Proprioception task, the objective is to pick a randomly placed cube on the table. The arm is randomly initialized, and the only observation available to the student is tactile forces and the last action. We observe that ASTEROID effectively solves the high-dimensional POMDP task with ($\sim 92\%$ success rate) by implicitly maintaining a belief over the underlying state. We plot the performance over iterations and the resulting behavior in Figure 13b. Note that in the tactile-pick (and some other) tasks, BC is also history-conditioned and achieves non-trivial performance (here, $\sim 48\%$) because certain evaluation reset conditions do not require significant or directed exploration to locate the object.



(a) Fixed Maze Navigation (b) Tactile Pick w/o Proprioception (c) Habitat Object Search

Figure 13: ASTEROID learns to solve POMDPs such as maze navigation, robot manipulation, and Habitat object search, where agents must act under limited fields of view, random resets, and partial sensing.

B.2 Robustness of ASTEROID to noise in privileged expert supervision

In Figure 14 (left), we evaluate the robustness of ASTEROID through an ablation study in which a noisy expert supervises the student. We inject Gaussian noise into the expert actions during training ($\sigma = 0.2$) and find that ASTEROID remains performant under noisy supervision. The slight performance improvement may stem from the regularizing effect of learning from noisy targets.

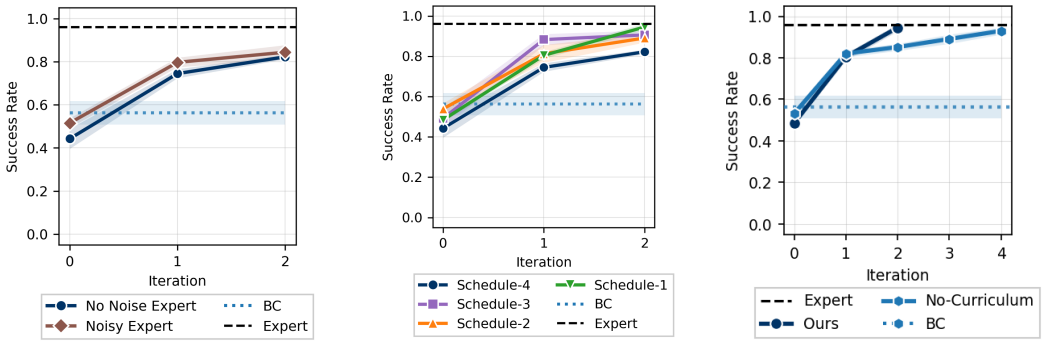


Figure 14: We run ablations on different parts of ASTEROID in the Tactile-Pick task. Left: we run ASTEROID with a noisy expert and observe no performance drop. Middle: we sweep over different curricula with increasing exploration horizons and find minimal sensitivity. Right: we show that adding curriculum improves the efficiency of ASTEROID by ensuring that as the horizon increases over iterations, the learned policy does not suffer from OOD histories and generate random actions.

B.3 How does the curriculum of exploration affect ASTEROID training?

At each iteration $k < K$, the policy interacts with the environment for k episodes, thereby inducing an increasing curriculum on on-policy rollout horizons. As the context window gradually expands, it allows the student to use longer histories to rule out hidden states or task instances inconsistent with its observations and feedback (Appendix B.5). The curriculum also avoids a subtle distribution-shift issue. Suppose the maximum context length is K, and we roll out an intermediate policy π_{θ_k} for more than k episodes. Since π_{θ_k} was only trained on contexts of length at most k, maintaining an in-distribution context would require a sliding window over the most recent k episodes. However, this window can discard early exploration outcomes. If a failed goal is no longer in context, the policy may re-explore it, producing redundant exploration histories that a fully converged policy would not generate. Informally, this reduces the effective number of length-H exploration contexts from $|\mathcal{Z}|^H$ arbitrary failure sequences to at most $\frac{|\mathcal{Z}|!}{(|\mathcal{Z}|-H)!}$ ordered non-redundant histories. This further reduces the effective context space, thereby making ASTEROID sample-efficient.

In Figure 14 (right), we compare the performance of ASTEROID across two different schedules: (i) No-Curriculum: the exploration horizon is sampled from a fixed range across all iterations, and (ii) Curriculum: the exploration horizon is sampled from a range that increases every iteration. We observe that curriculum-based exploration is more efficient than non-curriculum baselines, as it keeps student policy rollouts in distribution and makes on-policy data generation more sample-efficient. Furthermore, in Figure 14 (middle), we compare the robustness of the method across different values of increasing curriculum and observe that it is relatively insensitive to this parameter.

B.4 ASTEROID is more scalable than RL in complex domains.

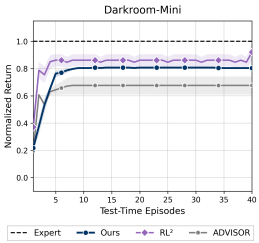


Figure 15: RL is effective over short horizons, but ASTEROID uses supervised updates to be effective across long horizons as well.

In Figures 6 and 8, we observe that RL-based baselines (RL^2, BC+RL^2, and ADVISOR) severely underperform in some long-horizon sparse reward tasks. To further ablate this, we run the RL baselines and our method in a smaller version of the Darkroom environment, where the grid size is 5x5. We observe that RL baselines learn competitive search policies in these small-scale environments. ASTEROID uses supervised learning to model expert actions while building exploration context via student rollouts, rather than relying on expert supervision and dynamic programming over random contexts to learn how to explore and exploit information. As a result, ASTEROID is more performant in high-dimensional long-horizon settings over RL baselines.

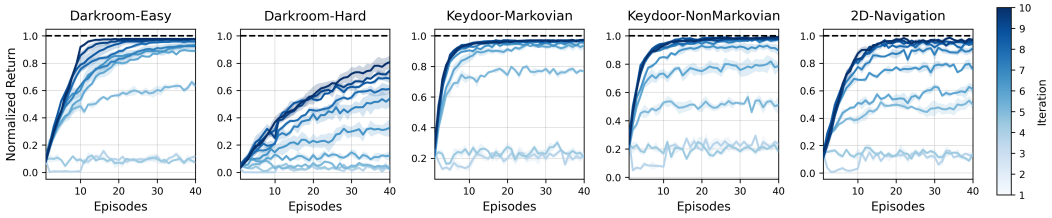


Figure 16: The test-time efficiency of ASTEROID increases significantly with more iterations of asymmetric on-policy distillation, demonstrating that as the context dataset grows, the policy can adapt over longer horizons, making exploration more efficient.

B.5 Does iterative context accumulation actually help ASTEROID?

Figure 13c shows exploration performance as a function of the number of in-context exploration iterations on a high-dimensional navigation task. In Habitat Navigation, performance improves steadily as additional exploration steps allow the policy to gather informative observations and refine its behavior, despite the complexity of the observations, almost converging to the posterior-sampling upper bound. Further, in Figure 16, we observe that the test-time efficiency of ASTEROID improves significantly with more iterations. These results indicate that ASTEROID effectively uses in-context interaction to accumulate task-relevant information, translating additional exploration into improved downstream task performance in realistic, partially observable environments, such as higher success rates in embodied object search in Habitat and in robot manipulation tasks. In Appendix B.3, we further include ablations showing the benefits and robustness of ASTEROID to the curriculum over exploration horizon across multiple iterations.

B.6 How well does ASTEROID generalize under shifted latent priors?

In Figure 17, we additionally measure how far this generalisation extends. In the 2D Navigation task, we sample training goals on a semicircle of radius 0.7 from the origin, and at evaluation we sample goals at increasing radii from 0.7 (in-distribution) to 1.0 (far out-of-distribution), keeping the dynamics and observations unchanged. This isolates the shift to the latent (goal) distribution alone. As shown in Figure 17, ASTEROID retains near-optimal performance near the training distribution (normalised return of 0.99 at radius 0.7 and 0.98 at 0.75) and degrades gradually as the goals move farther away (0.41 at radius 1.0). Generally, amortised POMDP solvers (Lee et al., 2023; Osband et al., 2013; Finn et al., 2017) that are learned via interaction or offline datasets suffer from a decline in performance as the train and test distributions diverge, and this observation is consistent with past results (Nikulin et al., 2025; Yu et al., 2020).

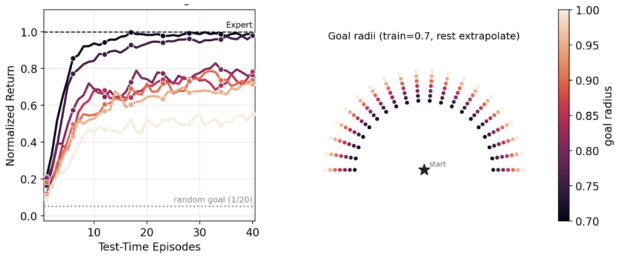


Figure 17: Generalization under shifted latent priors in 2D Navigation. Training goals lie at radius 0.7; evaluation goals are sampled at increasing radii up to 1.0. Performance degrades gradually with shift magnitude.

C Qualitative Results

C.1 ASTEROID learns efficient exploration in robot manipulation under limited sensing, and Navigation.

In Figures 18 and 19, we further compare the qualitative exploration behavior across the peg insertion and the navigation task. We observe that ASTEROID enables active exploration-like behavior during manipulation, in which the robot arm moves efficiently across the table to locate the peg and the target. Similarly, in the 2D navigation maze task, ASTEROID learns a policy that adaptively searches

for the goal in unexplored regions and converges to high reward in subsequent episodes once it can identify the goal region from past trials. We include full videos of the policies on our website: https://sites.google.com/view/icl-exploration, which also includes goal-directed exploration across different rooms in Habitat.

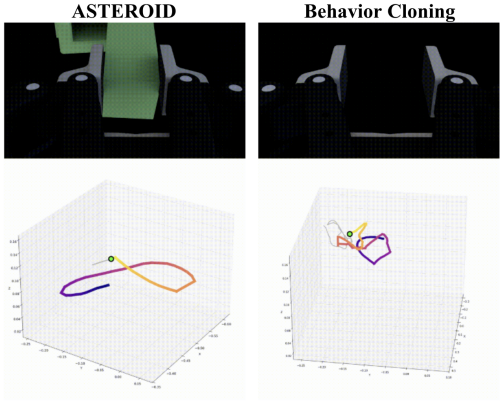


Figure 18: Robot positions (bottom row) during exploration for peg-insertion from wrist camera images (top row). ASTEROID adapts via contexts, while BC fails to adapt.

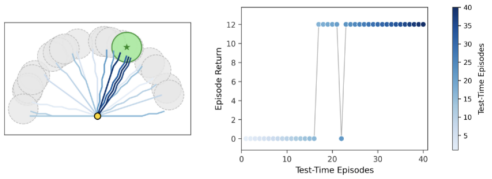
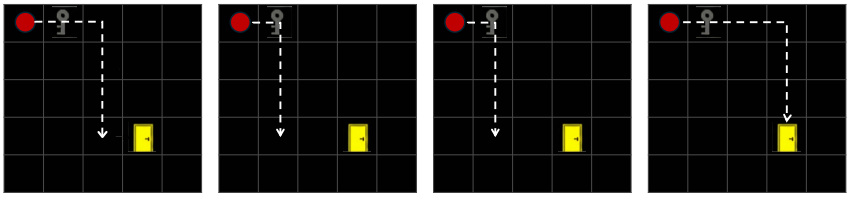
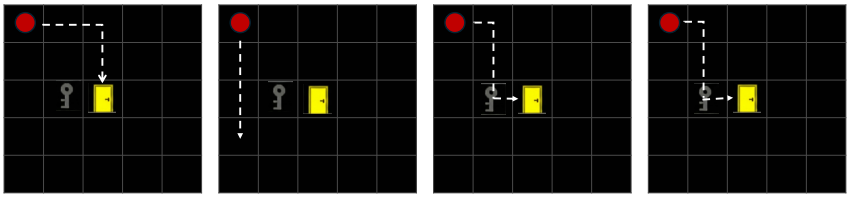


Figure 19: In 2D navigation, ASTEROID adapts its exploration based on context, converging to the true goal.

C.2 ASTEROID PRESENTS EMERGENT IN-CONTEXT PROPERTIES SUCH AS MEMORY AS A RESULT OF ON-POLICY IN-CONTEXT LEARNING.



(a) The agent finds the key in earlier episodes, can reference it back in its context, and always picks up the key before searching for the door.



(b) The agent finds the door in the first step but gets no reward. So, in the next episodes, it looks for the key and memorizes the door location. As a result, in episode 3, it finds the key and goes directly to the door (without having to search for it, since it has memory).

Figure 20: In the Key2Door task, the agent explores to find both the key and the door to get +1 reward for each task. ASTEROID learns an effective strategy: if it reaches the key or door first, it uses its context to remember the location, and upon finding the other goal, it can immediately complete the task without additional exploration.

To investigate whether the policy learned with ASTEROID actually attends to context, we construct a variant of the Key-Door environment in which the agent observes two boolean flags in its state whenever it lands on the key or the door during exploration. Policies trained with ASTEROID exhibit emergent in-context memory. If the agent discovered the key or door in earlier attempts, it can later retrieve that information from context and use it efficiently.

D PER-STEP INTERACTIVE IMITATION LEARNING FAILS IN HISTORY-CONDITIONED ASYMMETRIC DISTILLATION.



An extension to improve naive teacher-student distillation is to apply standard interactive learning methods, such as DAgger (Ross et al., 2011), on a history-conditioned student policy $\pi_\theta(\cdot|o_t, h_t)$. However, these interactive methods are not feedback-driven, i.e., they aggregate privileged action labels across observation and action histories, while the information needed for exploration often arrives only through rewards. Further, in POMDPs, DAgger surprisingly converges to an exploration strategy that is generally worse than naive distillation. Consider that a policy has to take H steps towards a potential goal to receive a reward and disambiguate between them. Here, $\pi_\theta(\cdot|o_t, h_t)$ cloned on expert behavior would randomly pick a door and generate an H step trajectory. Applying DAgger, an expert provides supervision at each step of the trajectory. In the figure above, we see that under different latent settings, the same student trajectory receives multi-modal action targets at each step. Supervised learning on this aggregated dataset would instead learn a policy that chooses a random goal at each step, which does not correspond to meaningful, effective exploration. Over repeated interactions, the updated policy uniformly explores the environment to find the goal, leading to extremely high exploration budgets. Overall, DAgger underperforms as a baseline in partially observed domains, even more so than naive behavior cloning (see Figure 8). Instead, ASTEROID employs trajectory/context-level supervision, i.e., given the history h_t, we supervise either the full interaction trajectory (in BAMDPs) / partial trajectory from the takeover state (in POMDPs). This enforces the policy to execute goal-directed actions over a longer horizon (rather than uniform dithering), resulting in much more efficient exploration, even in high-dimensional continuous action spaces (see Figure 8).

E DIDACTIC EXAMPLE: ASTEROID LEARNS EFFICIENT POSTERIOR SAMPLING.

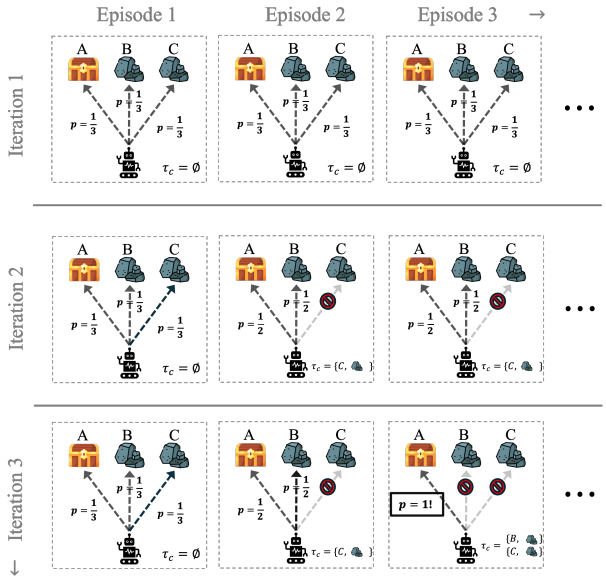


Figure 21: Toy Example demonstrating that ASTEROID learns adaptive in-context behavior, where it effectively explores based on history.

Consider a BAMDP with one hidden variable. For each task, a fixed goal $g \in \{A, B, C\}$ is sampled uniformly. The agent takes actions to move to one of the locations in $\{A, B, C\}$ and receives a reward of 1 if it matches g and 0 otherwise. A privileged teacher observes g and always chooses the correct

door, producing expert trajectories: $(A, 1)$, $(B, 1)$, or $(C, 1)$ depending on the goal. If the student is trained only on these expert rollouts with no context (i.e., $\tau^c = \emptyset$), then at deployment it cannot infer g and must choose uniformly, yielding expected reward $1/3$.

Dry run. At iteration $k = 0$, we train on expert-only trajectories with $\tau^c = \emptyset$, so the student has no information about g at decision time. The resulting policy chooses A, B, C uniformly, giving an expected reward of $1/3$.

Now deploy this student and collect on-policy contexts. If the true goal is $g = C$, then across episodes the observed contexts are of the form $(A, 0)$, $(B, 0)$, or $(C, 1)$, depending on which door was tried. At iteration $k = 1$, ASTEROID pairs each such context with a privileged teacher trajectory generated under the same goal. For example, conditioning on $\tau^c = (A, 0)$ rules out $g = A$, so the teacher data for this context consists only of successful choices at the remaining doors, i.e., $(B, 1)$ or $(C, 1)$. After distillation, the updated student still chooses uniformly when $\tau^c = \emptyset$. Still, it adapts after one failed attempt: given $\tau^c = (A, 0)$ it chooses between $\{B, C\}$ uniformly, yielding expected reward $1/2$ on the next episode.

At iteration $k = 2$, the student generates longer contexts such as $(A, 0; B, 0)$, $(A, 0; C, 1)$, etc. A context like $(A, 0; B, 0)$ uniquely identifies $g = C$, so the teacher data for this context concentrates on choosing C. After distillation, the student learns the corresponding rule: once it has observed two failures, it deterministically selects the remaining door and receives a reward of 1. Figure 21 depicts this progression: as τ^c grows with on-policy interaction, the student eliminates inconsistent goals based on context and its behavior approaches posterior sampling over $\{A, B, C\}$.

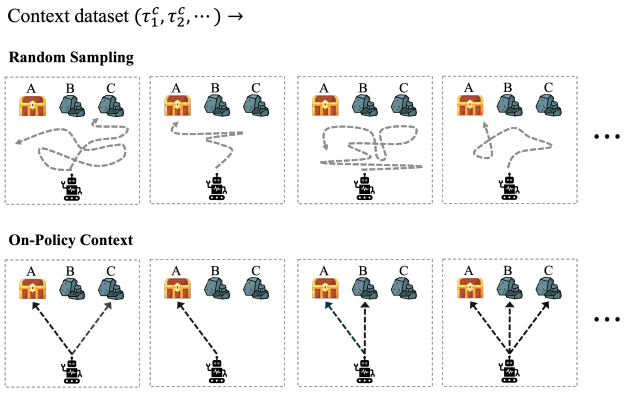


Figure 22: On-policy contexts ensure efficient coverage over relevant histories, while random exploration (Lee et al., 2023) can generate uninformative contextual information that harms training.

On-policy context enables more efficient learning than random exploration rollouts. In this example, a context τ_c is useful only if it induces a non-trivial posterior over the goal, i.e., it eliminates some candidates in $\{A, B, C\}$ based on observed rewards. Suppose instead that reaching each location requires multiple actions before any reward is revealed. Then a large fraction of randomly generated rollouts will fail to reach any door within the context horizon (row 1 in Figure 22), producing trajectories whose rewards are uninformative and whose posterior over g remains essentially unchanged. By contrast, on-policy contexts produced by the student concentrate on trajectories that actually reach candidate doors, since the student is trained on teacher trajectories that are goal-directed and therefore yield plausible search behavior (row 2 in Figure 22). As a result, the contexts in $\mathcal{D}_k$ more often contain the key feedback events that reduce uncertainty about g, providing a higher-signal training distribution for learning a context-conditioned policy.

F Training and Implementation Details

F.1 Training Details

Model Architecture. ASTEROID uses a causal transformer similar to Lee et al. (2023); Chen et al. (2021). Each token encodes the <observation, previous action, reward, terminal signal> tuple via an MLP encoder. For images, we encode observations with an ImageNet-pretrained ResNet18. For

discrete actions, we predict logits over the action space, and for continuous actions, we predict the mean and standard deviation of a Gaussian.

Asymmetric Sampling of datasets. For the robot manipulation tasks, to avoid overfitting, we skew the sampling ratio of the data collected across iterations to upweight the new data more. This significantly reduced overfitting to past data and enabled better policy learning.

Filtering for successful trajectories. As ASTEROID imitates privileged actions conditioned on context, we keep only the successful trajectories while collecting the datasets during each iteration. This ensures that the learned in-context policy samples from a distribution of expert actions that are optimal under the true parameters that could generate the given context.

Task Experts ASTEROID depends on a strong privileged expert but is agnostic to how it's generated: it may leverage simple planners, use RL, imitation learning, or hybrid methods (Schulman et al., 2017; Yin et al., 2026; Zhang et al., 2026). Concretely: for all navigation and gridworld tasks, the experts are shortest-path planners. For manipulation, we train state-based policies via large-scale simulation RL (Yin et al., 2026) with hyperparameters from the original implementations. Privileged state makes these experts cheap to train. E.g., in peg insertion, a state-based teacher (object and goal 6-D pose) reaches 90+% success in 8 hrs, while a wrist-camera policy takes 4+ days to reach only 10%. This state information is not available at deployment, and thus, we learn exploration via ASTEROID. Most of our baselines also use privileged experts, so teacher cost is not exclusive to our method.

Compute. DPT, ASTEROID, BC, and AAWR learn context-conditioned policies, and we use the same architecture and evaluation setup. Thus, these methods use the same tokens and computation during rollouts. During training, ASTEROID additionally conditions on the expert observations, adding at most |T| tokens more than DPT, where |T| is the length of an episode. All models were trained on a single L40 GPU; training takes 1–6 h, and the main bottleneck is environment rollouts, not training. For IsaacLab tasks, we used 2× RTX-4090s for our method and all baselines. We carefully match the compute in terms of environment interactions across all baselines (except when comparing efficiency). In all experiments, each baseline is trained on as much or more data than ASTEROID. ASTEROID runs K iterations with M environment steps each; all offline baselines (BC, DPT, SPOC, AAWR) train on a dataset of size $K \times M$. Figure 8 plots training steps vs. performance for the manipulation baselines. RL baselines are trained to convergence. In Table 1, we outline the parameters:

Table 1: Per-environment data and compute budget for ASTEROID and baselines.

Environment	K (iters)	ASTEROID data (trajs)	Offline baselines data	RL baselines env steps	Expert labels (rollouts)	Teacher cost
Gridworld (Darkroom, 2D-Nav)	10	$10 \times 1000 = 10$k (env horizon H)	10k expert trajs	to convergence	10k	negligible (planner)
Tactile-Pick	3	$3 \times 8192 \approx 24$k	≈24k expert trajs	$\sim 10^8$	≈24k	∼8 h (state-based RL)
Peg-Insertion	4	$4 \times 4096 \approx 16$k	≈16k expert trajs	$\sim 10^8$	≈16k	∼8 h (state-based RL)
Habitat	5	$5 \times 1024 = 5120$	5120 (Distilled Planner)	n/a	5120	negligible (planner)

F.2 TASKS

Wrist-Camera Peg Insertion. (Yin et al., 2026) A 7-DoF robot must locate a peg on a tabletop and insert it into the corresponding hole using only a wrist-mounted RGB camera and proprioception. This is a single-episode POMDP: the peg and arm are randomly reset at the start of each episode, and the robot must search under a limited camera field of view. We denote the privileged state as $s_t = (q_t, p_{\text{peg}}, p_{\text{hole}})$, where q_t is the robot proprioceptive state and $p_{\text{peg}}, p_{\text{hole}}$ are the peg and hole positions. The student observes $o_t = (I_t^{\text{wrist}}, q_t, a_{t-1})$, where I_t^{wrist} is the wrist-camera RGB image and a_{t-1} is the previous action. The reward indicates successful insertion, and expert trajectories are generated using large-scale RL as described in Yin et al. (2026). An effective exploration strategy must scan the workspace, localize the peg and hole from partial views, and then execute the contact-rich insertion behavior.

Tactile Pick. (Yin et al., 2026; Higuera et al., 2023) The robot must locate and pick up a cube using proprioceptive and tactile feedback, without visual observations. This is a single-episode POMDP: the cube pose is randomly reset and hidden, so the agent can only infer it through contact during the episode. We denote the privileged state as $s_t = (q_t, p_{\text{cube}})$, where q_t is the robot proprioceptive

state and p_{cube} is the cube position. The student observes $o_t = (q_t, \tau_t, a_{t-1})$, where τ_t is the 2×3-dimensional force feedback from the two fingers of the parallel-jaw gripper. The reward indicates successful pickup, and expert trajectories are generated using large-scale RL as described in Yin et al. (2026). An effective exploration strategy must probe the workspace, use contact events to infer the cube location, and then transition from search to grasping. We also evaluate a harder variant with observations $o_t = (\tau_t, a_{t-1})$, where the robot observes the world only through contact forces and past actions. In this setting, the robot starts from a random configuration and must use its interaction history to maintain a belief over both its own joint configuration and the cube position.

Dark Room. (Chevalier-Boisvert et al., 2023; Lee et al., 2023) The agent must find a hidden goal in a partially observable gridworld. We use three versions with different grid sizes: Mini (5×5), Easy (10×10), and Hard (25×25). This is a BAMDP: the agent interacts with the environment over multiple test-time episodes, the goal remains fixed across episodes, and the agent is reset to $(0, 0)$ at the start of each episode. We denote the privileged state as $s_t = (p_x, p_y, g_x, g_y)$, where (p_x, p_y) is the agent position and (g_x, g_y) is the hidden goal. The student observes $o_t = (p_x, p_y, r_t)$, where $r_t = 1\{(p_x, p_y) = (g_x, g_y)\}$. The expert is obtained by planning the shortest path to the hidden goal using privileged goal information. An effective exploration strategy must use the interaction history to avoid revisiting locations that already produced zero reward, search new candidate goals, and exploit the goal once it is found. This is exactly the structure induced by ASTEROID: given a student history, if a rewarding position has been observed, the expert labels return to it; otherwise, the expert labels target positions not yet ruled out by the history. We train and evaluate on an 80–20 random split of goal locations.

Key2Door. (Chevalier-Boisvert et al., 2023; Lee et al., 2023) The agent must first locate a hidden key and then navigate to a door; the key is picked up and the door is opened by moving to their respective positions. This is a BAMDP with non-Markovian observations: the agent interacts across multiple test-time episodes, and optimal behavior depends on remembering which locations have been searched and whether the key has already been acquired. We denote the privileged state as $s_t = (p_x, p_y, k_x, k_y, d_x, d_y, 1\{\text{has key}\})$, where (p_x, p_y) is the agent position, (k_x, k_y) is the key position, (d_x, d_y) is the door position, and the final bit indicates whether the agent has the key. The student observes $o_t = (p_x, p_y, r_t)$. In the Markovian version, the agent receives a reward of $+1$ at every timestep after acquiring the key, effectively revealing the task stage. This dense reward also leads the RL baselines to get stuck in a local minimum and perform poorly. In the non-Markovian version, the agent receives a reward of $+1$ only at the timestep when it picks up the key, so it must use its history to decide whether to continue searching for the key or switch to searching for the door. In both versions, the agent receives an additional reward of $+1$ upon reaching the door with the key, and the episode ends. An effective exploration strategy must search for the key, remember how to acquire it, and then search for or navigate to the door.

Ant and 2D Navigation. (Rakelly et al., 2019; Zintgraf et al., 2021) The agent must discover and reach a hidden goal, either in a continuous 2D navigation domain or with a high-dimensional quadruped. These are BAMDPs: the goal remains fixed across multiple episodes within a trajectory, while the agent resets between episodes and must use in-context exploration to identify unvisited candidate goals. We denote the privileged state as $s_t = (x_t, g)$, where x_t is the agent state and g is the hidden goal. The student observes $o_t = (x_t, r_t)$, but not g. The reward is sparse, with $r_t = 1\{\|x_t - g\|_2 \leq \epsilon\}$ for a success threshold ϵ. The expert uses privileged access to g to produce goal-directed behavior. We train and evaluate on an 80–20 split of goal locations. An effective exploration strategy must use history to avoid revisiting unlikely goals, sample unexplored regions, and exploit the goal once discovered.

Habitat Object Navigation. (Puig et al., 2024) The agent must find a target object in a realistic indoor environment, where the object is placed at one of k possible locations. This is a partially observed task: the agent interacts for a single episode, while the hidden object location remains fixed throughout the episode and must be inferred from experience. We denote the privileged state as $s_t = (\mathcal{M}, x_t, g)$, where $\mathcal{M}$ is the map of the house, x_t is the robot pose, and $g \in \{1, \ldots, k\}$ is the hidden object location. The student observes $o_t = (x_t, I_t, b_t)$, where I_t is the onboard camera observation and $b_t \in \{0, 1\}$ is the output of an object detector applied to the current image. The reward is sparse and indicates whether the target object has been found. The expert is an oracle

planner that uses the map and object location to navigate to the target. The student policy must use history to eliminate previously checked locations, search among the remaining candidates, and stop once the detector identifies the object.

Procgen Maze Navigation. The agent must navigate a procedurally generated maze to find a hidden goal. This is a single-episode POMDP: the maze layout, initial position, and goal location are unknown, and the agent observes only a local 3×3 window centered at its current position. We denote the privileged state as $s_t = (\mathcal{M}, p_t, g)$, where $\mathcal{M}$ is the maze layout, p_t is the agent position, and g is the goal location. The student observes $o_t = \mathcal{M}_{3\times3}(p_t)$, the local egocentric crop centered on the agent, but not the global map or the goal. The reward is sparse and indicates whether the agent reaches the goal. The expert plans the shortest path using Q-learning and privileged access to the full maze layout and goal location. An effective exploration strategy must use temporal context to build an implicit belief over the maze, avoid repeatedly visiting uninformative regions, and search until the goal is found. In Fixed Maze Navigation, we evaluate a simplified variant with a fixed maze layout but random initial and goal positions, isolating the challenge of goal-directed exploration under local 3×3 observations from the additional generalization required by procedurally generated layouts.

Real Robot Manipulation. We deploy on a UR5e arm with a parallel-jaw gripper. The cube is placed at a random position within a 10cm $\times$ 2cm region on the table, and the arm starts from a fixed initial configuration. The student observes only proprioception, i.e., joint angles and gripper jaw position, with no cameras or touch sensing, so the cube location can only be inferred from whether a grasp attempt closes on the object. Here, if the robot attempts a grasp at a false location, the gripper closes all the way, eliminating that location. We train the privileged expert in IsaacLab with large-scale RL and domain randomisation over physics parameters following Mittal et al. (2023); Yin et al. (2026). We train both ASTEROID and the BC baseline in simulation with the same partial observations, then transfer them to hardware. We evaluate each method on 10 trials with the same set of cube positions, and count a trial as successful if the cube is lifted within 20 seconds.

G Baseline Implementation Details

DPT (Lee et al., 2023). We use the official implementation of DPT with the default hyperparameters from Lee et al. (2023). For a fair comparison, we set the dataset size to match the total data collected over all iterations of ASTEROID.

SPOC (Ehsani et al., 2023). SPOC augments naive distillation with sampled positional embeddings. We implement SPOC within our codebase and train it on the same total amount of data used across all iterations of ASTEROID.

RL baselines. For continuous-control RL baselines, we use high-quality implementations from RSL-RL (Mittal et al., 2023) and Omni-Reset (Yin et al., 2026), which support large-scale GPU-parallel training. We implement ADVISOR in RSL-RL and tune its α and β parameters over the ranges reported in the original paper. For the meta-RL baselines, we use RL^2/PPO baselines (Duan et al., 2016) and adapted versions that incorporate privileged expert supervision, thereby reducing the computational burden of learning exploration and task execution from sparse rewards alone. For gridworld tasks, we use Stable-Baselines3 PPO implementations with task-specific modifications. On the wrist-camera task, we run the vision-based RL baseline for two seeds due to computational cost: 10^8 environment steps take roughly four days on an RTX-enabled GPU. Despite this scale, the learned exploration policy achieves only 10% success.

AAWR (Hu et al., 2025). We use an extended version of the original AAWR implementation. To construct the initial offline dataset, we add Gaussian noise to generate broad coverage over possible histories. As shown in Figure 11, AAWR is sensitive to this initial dataset coverage, whereas ASTEROID avoids this dependence by collecting context on-policy. AAWR and SPOC were not originally designed for BAMDPs, but we include them in both settings for completeness.

Distilled Planner. For Habitat object search, we construct an upper-bound baseline by distilling a posterior-sampling planner under uniform prior. The dataset consists of expert trajectories that visit the possible object locations in a fixed order and terminate upon finding the target. This produces

goal-directed exploration behavior, which we distill into a transformer policy. Unlike ASTEROID, however, this baseline assumes access to the full set of possible object locations and the underlying task structure during training, which is unrealistic.

G.1 PSEUDOCODE

Algorithm 1 ASTEROID

Require: iterations K, tasks per iteration N, horizon H, privileged teacher π^*

Initialize $\mathcal{D}_0 \leftarrow$ null context expert trajectories, student π_{θ_0} trained on D_0
for $k = 1, \ldots, K$ **do**
  $\mathcal{D}_k \leftarrow \mathcal{D}_{k-1}$
  `// Collect on-policy contexts and privileged labels`
  for $i = 1, \ldots, N$ **do**
    Sample $z \sim p(z)$ and instantiate M_z
    Roll out $\pi_{\theta_{k-1}}$ in M_z for k episodes: $h_k^{(i)} = (\tau^0, \ldots, \tau^{k-1})$
    Roll out π^* in M_z for one episode: $\tau^{*,(i)} = (o_0^*, a_0^*, r_0^*, \ldots, o_{H-1}^*, a_{H-1}^*, r_{H-1}^*)$
    Add labels $\mathcal{D}_k \leftarrow \mathcal{D}_k \cup \{(h_k^{(i)}, o_{0:t}^*, r_{0:t-1}^*, a_t^*)\}_{t=0}^{H-1}$
  end for
  Train π_{θ_k} on $\mathcal{D}_k$ using Eq. 1 `// Update student by conditional imitation`
end for
return π_{θ_K}

Algorithm 2 ASTEROID for general POMDPs

Require: iterations K, samples per iteration N, horizon H, privileged teacher π^*

Initialize $\mathcal{D}_0 \leftarrow$ null context expert trajectories, student π_{θ_0} trained on D_0
for $k = 1, \ldots, K$ **do**
  $\mathcal{D}_k \leftarrow \mathcal{D}_{k-1}$
  `// Collect on-policy contexts and privileged labels`
  for $i = 1, \ldots, N$ **do**
    Sample t uniformly in $[1, H-1]$ `// or based on the sampling curriculum`
    Roll out $\pi_{\theta_{k-1}}$ in POMDP for t steps: $h_k^{(i)} = (o_0, a_0, r_0 \cdots o_{t-1}, a_{t-1}, r_{t-1})$
    Roll out π^* in the same episode from o_t: $\tau^{*,(i)} = (o_t^*, a_t^*, r_t^*, \ldots, o_{H-1}^*, a_{H-1}^*, r_{H-1}^*)$
    Add labels $\mathcal{D}_k \leftarrow \mathcal{D}_k \cup \{(h_k^{(i)}, o_{t:t'}^*, r_{t:t'-1}^*, a_t^{*\prime})\}_{t'=t}^{H-1}$
  end for
  `// Update student by conditional imitation`
  Train π_{θ_k} on $\mathcal{D}_k$ using Eq. 1
end for
return π_{θ_K}

Algorithm 3 ASTEROID Deployment

Require: trained student π_{θ_K}, horizon T

Initialize context $h \leftarrow \emptyset$
`// Build context from online interaction`
for $t = 0, \ldots, T-1$ **do**
  Observe o_t, sample $a_t \sim \pi_{\theta_K}(\cdot \mid h, o_t)$
  Execute a_t, receive r_t
  Update context $h \leftarrow (h, o_t, a_t, r_t)$
end for

G.2 HYPERPARAMETERS

Table 2: Hyperparameters for ASTEROID on Gridworld tasks.

Hyperparameter	**Value**
Training	
Dataset size (per iteration)	1000
ASTEROID iterations	10
Batch size	64
Learning rate	1×10^{-4}
Weight decay	1×10^{-4}
Number of training epochs (per iteration)	100
Warmup ratio	0.03
Gradient clipping	1.0
Optimizer	AdamW
LR scheduler	Warmup + Cosine Decay
Model Architecture (Decision Transformer (Chen et al., 2021))	
Architecture	GPT-2Model
Tokenizer	Project (o_t, a_{t-1}, r_{t-1}) linearly
Parameters	$\sim$3M
Number of layers	4
Number of attention heads	4
Embedding dimension	256
Dropout	0.1
Max Context length	$H \times K$
Evaluation	
Test-time episodes	40
Episode length	100

Table 3: Hyperparameters for ASTEROID on IsaacLab tasks. The exploration horizon denotes the $[\min, \max]$ fraction of the episode during which the exploration policy acts before handing off to the expert.

Hyperparameter	**Value**
Training	
Demonstrations per iteration	8192 for Tactile-Cube-Pick / 4096 for Wrist-Peg-Insertion
Training steps per iteration	50,000 gradient steps
Learning Rate	1×10^{-4}
Optimizer	AdamW
Curriculum Schedule (Ratio of episode length)	
Exploration horizon (iteration 1)	$[0.2, 0.5]$
Exploration horizon (iteration 2)	$[0.3, 0.8]$
Exploration horizon (iteration 3)	$[0.4, 0.9]$
Evaluation	
Evaluation episodes	64
Episode length	240

H Additional Theory

We present a theoretical analysis of properties of learning in-context exploration using ASTEROID. We discuss why on-policy contexts help: since ASTEROID labels the student's own histories, it behaves like DAgger (Ross et al., 2011) in history space, while an offline in-context learner behaves like behaviour cloning. We analyse the BAMDP setting of Algorithm 1. Reusing notation from Section 3, h denotes the student's history, d_π denotes the history distribution induced by a policy π,

$d_{\mathcal{D}}$ denotes the empirical history distribution in a dataset, $\pi^*(a \mid o, z)$ denotes the privileged teacher, and $\pi_\theta(a \mid h)$ denotes the history-conditioned student. We write $\pi_k \equiv \pi_{\theta_k}$ for the student iterate at ASTEROID iteration k.

Theorem. Under compliance, on-policy coverage, and bounded episode cost, ASTEROID approximately recovers posterior sampling on the histories it visits. Given a supervised learner with average error ϵ on its own histories, its gap to posterior sampling over K episodes is $O(\epsilon K)$, whereas an offline learner with the same training error can accumulate $O(\epsilon K^2)$. The total regret is $\mathrm{Reg}_{\mathrm{PS}}(K) + O(\epsilon K)$.

We present formal statements and proofs below. The key step is that privileged labels queried on the student's own histories are exactly posterior-sampling labels (Lemma 1). In addition, we show that an offline in-context learner with the same training error can incur $O(\epsilon K^2)$ error (Lemma 2), because the test-time histories it encounters drift away from its fixed training distribution.

H.1 Preliminaries

Following PSRL (Osband et al., 2013), we analyse ASTEROID at the episode level: at the start of each episode, the policy selects a hypothesis $\tilde{z}$ and executes the teacher $\pi^*(\cdot \mid \cdot, \tilde{z})$ for the full episode. Here, the student induces an episode-level policy $\pi(\tilde{z} \mid h)$. We define the posterior-sampling policy

$$\pi^{\mathrm{PS}}(\tilde{z} \mid h) := p(\tilde{z} \mid h),$$

which samples a hypothesis from the posterior and commits to it for the episode. This produces a policy $\pi^{\mathrm{PS}}(a|o,h) = \int \pi^*(a|o,z)\ \mathrm{p}(z|h)\ dz$. Here, we evaluate the policy over K online episodes of fixed episode horizon H. Per-episode returns are normalised to $[0, 1]$.

Let d_k^π denote the distribution of context at the start of episode k when rolling out π: the history h contains the first $k-1$ episodes. We measure the student's error in episode k by its deviation from posterior sampling on its own contexts,

$$e_k(\pi) := \mathbb{E}_{h \sim d_k^\pi}\left[\mathrm{TV}\big(\pi(\cdot \mid h),\ \pi^{\mathrm{PS}}(\cdot \mid h)\big)\right]. \tag{2}$$

Minimising equation 1 controls this via Pinsker's inequality.

Assumption 1 (Compliance)**.** *The policy used to generate context is unprivileged: at time t, its action depends only on the observed history, not directly on the latent variable z.*

This mirrors Assumption 1 of DPT (Lee et al., 2023).

Assumption 2 (On-policy support coverage)**.** *At ASTEROID iteration k, the aggregated dataset covers the support of the histories visited by the current rollout policy up to context of k episodes.*

Assumption 3 (Realizability and exact optimization)**.** *The policy class Π contains π^{PS}, and at every iteration the conditional maximum-likelihood objective in equation 1 is minimised exactly in the population limit.*

Assumption 4 (Bounded episode cost)**.** *A wrong hypothesis in episode k affects the return only through that episode, i.e., the episode-k return gap between π and π^{PS} is at most $e_k(\pi)$. Summing over episodes gives $J(\pi^{\mathrm{PS}}) - J(\pi) \le \sum_{k=1}^{K} e_k(\pi)$.*

Remark 1. *The student π_θ acts only on its observed history and never on z, so every context generated by ASTEROID rollouts satisfies Assumption 1 by construction.*

H.2 Privileged Labels Provide Posterior-Sampling Supervision

The key observation is that privileged labels are posterior-sampling labels when conditioned on compliant histories. A history is compliant if it is generated by a policy that does not directly observe the latent variable z, as in prior work on learning a posterior from in-context supervision (Lee et al., 2023). Intuitively, compliance ensures that the history reveals information about z only through observations, actions, and rewards available to the student, not through hidden privileged information.

Lemma 1 (Privileged labels equal posterior-sampling labels)**.** *Let h be a compliant history and o a query observation. If the privileged teacher label is sampled as $a^* \sim \pi^*(\cdot \mid o, z)$, then*

$$\mathbb{P}(a^* = a \mid h, o) = \pi^{\mathrm{PS}}(a \mid h, o).$$

This lemma says that we do not need to explicitly infer z: if we train on privileged labels conditioned on compliant histories, maximum likelihood directly trains the posterior-predictive policy. The proof follows Lee et al. (2023).

Theorem 1 (ASTEROID approximates posterior sampling on covered histories)**.** *Under Assumptions 1–3, at every iteration k, for every context h in the support of the aggregated on-policy dataset $\mathcal{D}_k$,*

$$\pi_k(\cdot \mid h) = \pi^{\mathrm{PS}}(\cdot \mid h).$$

Consequently, as the context horizon reaches the full online interaction horizon, ASTEROID matches posterior sampling on the covered trajectory support.

Proof sketch. The proof is by induction over iterations.

Base case. At iteration 0, ASTEROID trains on null-context expert data. For the empty context $h = \emptyset$ (and along the expert query prefixes, per Remark 1), Lemma 1 implies that the conditional label distribution is exactly $\pi^{\mathrm{PS}}(\cdot \mid h)$. Minimizing cross-entropy over $\mathcal{D}_0$ under Assumption 3 recovers this conditional distribution, so $\pi_0(\cdot \mid h) = \pi^{\mathrm{PS}}(\cdot \mid h)$ on the support of the dataset.

Inductive step. Assume that at iteration k, $\pi_{k-1}(\cdot \mid h) = \pi^{\mathrm{PS}}(\cdot \mid h)$ for every context h of depth at most $k-1$ episodes in the covered support. ASTEROID now rolls out π_{k-1} for k episodes to collect contexts. By Remark 1 these contexts are compliant, and since the two policies agree up to that depth, the collected histories follow the same distribution as those posterior sampling would generate up to that horizon. At each such history h, ASTEROID queries the privileged teacher. By Lemma 1, the conditional label distribution given h is $\pi^{\mathrm{PS}}(\cdot \mid h)$, so exact minimization of equation 1 yields $\pi_k(\cdot \mid h) = \pi^{\mathrm{PS}}(\cdot \mid h)$. Because the context schedule expands from $k-1$ to k episodes, the agreement extends to the larger support, proving the inductive claim. □

Theorem 1 is an on-support statement: ASTEROID does not need offline coverage of all possible histories. It constructs the relevant support by rolling out the student and querying the teacher on the histories the student actually visits.

H.3 Offline versus On-Policy Training

DPT (Lee et al., 2023) recovers posterior sampling in the population limit, and Theorem 1 shows that this holds under on-policy support. We now relax Assumption 3. The argument follows the classic comparison of behavior cloning and DAgger (Ross et al., 2011), with states replaced by histories and actions by $\tilde{z}$.

Offline error compounds across episodes. Suppose an offline learner is trained on contexts from a fixed distribution d_k^{off}, and has small error there, $e_k^{\mathrm{off}}(\pi) \le \epsilon$. At deployment, however, it is evaluated on its own contexts d_k^{π}. Since TV is bounded,

$$e_k(\pi) \le \epsilon + \mathrm{TV}(d_k^{\pi}, d_k^{\mathrm{off}}).$$

Lemma 2 (Offline history mismatch compounds)**.** *If the offline contexts are generated by posterior sampling, $d_k^{\mathrm{off}} = d_k^{\mathrm{PS}}$, then $\mathrm{TV}(d_k^{\pi}, d_k^{\mathrm{off}}) \le (k-1)\epsilon$, and hence $e_k(\pi) \le k\epsilon$.*

Proof sketch. Couple a rollout from the offline context distribution with a rollout from the learner. In each earlier episode, the learner makes an error with probability at most ϵ. Once such an error occurs, the subsequent history may differ from the offline history, since future observations, rewards, and actions are conditioned on the previous trajectory. A union bound over the first $k-1$ episodes bounds the probability that the two histories have diverged by $(k-1)\epsilon$, which upper-bounds the total variation distance. Substituting into the previous inequality gives $e_k(\pi) \le k\epsilon$. □

For other offline distributions, such as random contexts, the mismatch is only larger. Summing over K online episodes,

$$\sum_{k=1}^{K} e_k(\pi) \le \epsilon \sum_{k=1}^{K} k = O(\epsilon K^2).$$

Thus, even if the offline learner has small error on its dataset, its error can compound quadratically because the histories it sees at deployment drift away from the histories used for training.

On-policy error is linear in the number of episodes. ASTEROID avoids this mismatch by training on the student's own histories. Here, we assume the number of training iterations equals the number of online episodes, so we index both by K. Let $\pi_1, \ldots, \pi_K$ be the policies produced during iterative training, and define the on-policy loss

$$\ell_i(\pi) := \frac{1}{K} \sum_{k=1}^{K} e_k^{(i)}(\pi),$$

where $e_k^{(i)}(\pi)$ denotes the per-episode deviation of equation 2 evaluated on the histories induced by the iterate π_i. Let $\epsilon_K := \min_{\pi \in \Pi} \frac{1}{K} \sum_{i=1}^{K} \ell_i(\pi)$ be the best average error achievable within the policy class, and suppose the supervised learner is no-regret with online regret γ_K:

$$\frac{1}{K} \sum_{i=1}^{K} \ell_i(\pi_i) \le \epsilon_K + \gamma_K.$$

Lemma 3 (On-policy error)**.** *There exists an iterate $\hat{\pi} = \pi_{\hat{\imath}} \in \{\pi_1, \ldots, \pi_K\}$ such that*

$$\ell_{\hat{\imath}}(\hat{\pi}) \le \epsilon_K + \gamma_K,$$

i.e., its average error on its own induced history distribution is at most $\epsilon_K + \gamma_K$.

Proof. The average on-policy loss of the iterates is at most $\epsilon_K + \gamma_K$. Therefore, at least one iterate has loss no larger than this average. □

Theorem 2 (On-policy error is linear in the number of episodes)**.** *Let $\hat{\pi}$ be the iterate from Lemma 3, and $\epsilon := \epsilon_K + \gamma_K$. Then the cumulative error over K online episodes satisfies*

$$\sum_{k=1}^{K} e_k(\hat{\pi}) = K\, \ell_{\hat{\imath}}(\hat{\pi}) \le K(\epsilon_K + \gamma_K) = O(\epsilon K).$$

Proof sketch. Unlike the offline learner, ASTEROID does not train on histories from a fixed behavior distribution: at each iteration, the histories in the dataset are generated by the current student policy itself. On-policy histories are collected and relabeled at the next iteration, so the student learns to continue from its own histories. When the learned policy is evaluated on its own histories, there is therefore no additional mismatch term of the form $\mathrm{TV}(d_k^{\hat{\pi}}, d_k^{\mathrm{off}})$. The only remaining error is the supervised error on the student's own induced history distribution, which Lemma 3 bounds by $\epsilon_K + \gamma_K$ on average over context depths. Multiplying by K gives the claim. Thus, on-policy context aggregation removes the history-mismatch term that makes the offline error grow like $O(k\epsilon)$ per episode: the cumulative error is linear rather than quadratic in the number of online episodes. □

Corollary 1 (Total regret)**.** *$\mathrm{Reg}_{\mathrm{PS}}(K)$ is the Bayesian regret of posterior sampling over K episodes. By Assumption 4 and Theorem 2, $J(\pi^{\mathrm{PS}}) - J(\hat{\pi}) \le \epsilon K$, so the Bayesian regret of ASTEROID is*

$$\mathrm{Reg}_{\mathrm{PS}}(K) + O(\epsilon K),$$

posterior sampling's regret plus a linear approximation term.

Corollary 2 (Finite samples)**.** *By Lemma 1, each iteration is a standard conditional maximum-likelihood problem. Applying the finite-class MLE guarantee of FLAMBE (Agarwal et al., 2020, Theorem 21) with N on-policy contexts per iteration (and a union bound over the K iterations), with probability at least $1 - \delta$,*

$$\sum_{k \le K} e_k(\hat{\pi}) = \widetilde{O}\Big(K\big(\epsilon + \sqrt{\log(K|\Pi|/\delta)/N}\big)\Big),$$

so finite data adds a $\widetilde{O}(K/\sqrt{N})$ term without changing the linear dependence on the number of episodes.