Learning to Explore in POMDPs with Informational Rewards

Annie Xie [1] Logan Mondal Bhamidipaty [1] Evan Zheran Liu [2] Joey Hong [3] Sergey Levine [3] Chelsea Finn [1]

Abstract

Standard exploration methods typically rely on random coverage of the state space or coverage-promoting exploration bonuses. However, in partially observed settings, the biggest exploration challenge is often posed by the need to discover information-gathering strategies—e.g., an agent that has to navigate to a location in traffic might learn to first check traffic conditions and then choose a route. In this work, we design a POMDP agent that gathers information about the hidden state, using ideas from the meta-exploration literature. Our approach provides an exploration bonus that rewards the agent for gathering information about the state that is relevant for completing the task. While this requires the agent to know what this information is during training, it can obtained in several ways: in the most general case, off-policy algorithms can leverage knowledge about the entire trajectory to determine such information in hindsight, but the user can also provide prior knowledge (e.g., privileged information) to help inform the training process. Through experiments in several partially-observed environments, we find that our approach is competitive with prior methods when minimal exploration is needed, but substantially outperforms them when more complex strategies are required. Our algorithm also shows the ability to learn without any privileged information, by reasoning about the entire trajectory in hindsight and and effectively using any information it reveals about the hidden state.

1. Introduction

In many realistic decision-making problems, the agent lacks full information about the environment state, which makes learning to act optimally challenging. For example, learning to drive to work in the shortest amount of time is difficult if information about traffic and road blockages is unknown. A common way to simplify learning under such partial observability is to provide the learner with privileged information at training time, such as the full underlying states. Prior approaches leverage this privileged information to train an omniscient expert, which then supervises a policy without such information (Levine et al., 2016; Pan et al., 2017; Pinto et al., 2017; Chen et al., 2020; Baisero & Amato, 2021; Lambrechts et al., 2023). While the expert supervision is useful for *solving* the task, it leaves a critical aspect unaddressed: how to *explore* and actually gather the information used by the expert.

To better understand this issue, consider the naïve approach of training a policy without privileged information to directly *imitate* an expert with access to such information. Applied to our example of efficiently driving to work, an omniscient person with knowledge of the road and traffic conditions can directly take the optimal route. However, people do not typically know the exact conditions *a priori* and instead must initially rely on traffic reports and observable cues on the road to make their decision. In other words, the optimal policy without omniscient information is gather information, and simply imitating the omniscient expert does not yield this exploration or information-gathering strategy. Initial attempts to address this issue combine expert imitation with a standard reinforcement learning (RL) objective to maximize the reward in the hopes of learning behaviors beyond those of the privileged expert (Baisero & Amato, 2021; Weihs et al., 2021; Nguyen et al., 2022; Walsman et al., 2022; Shenfeld et al., 2023; Wang et al., 2023). However, such approaches can struggle because learning complex information-gathering strategies from the generic RL objective can be challenging.

More generally, optimally behaving in a partially-observable Markov decision process (POMDP) (Kaelbling et al., 1998) requires both *exploiting* known information to solve the task, and *exploring* to reduce uncertainty; and many existing POMDP methods that leverage privileged information do not address the latter. On the other hand, the meta-RL literature has carefully studied such exploration for a special case of POMDPs, known as hidden parameter Markov decision processes (Doshi-Velez & Konidaris, 2016), where the unobserved component of the state is fixed throughout an

[1]Stanford University [2]Imbue [3]UC Berkeley. Correspondence to: Annie Xie <anniexie@stanford.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

episode. These approaches induce effective exploration with objectives that incentivize gathering information that help infer the unobserved state. In our work, we aim to extend these ideas to the general POMDP setting, where the unobserved state may change at any point. Whereas meta-RL solutions can first explore to gather all necessary information and then use this information solve the task (Liu et al., 2021; Norman & Clune, 2023) because the unobserved state does not change within an episode, general POMDPs require gathering new information whenever task-relevant changes occur in the unobserved state. Hence, we aim to design an agent that decides *when* to explore and effectively interleaves exploration with exploitation.

From a high level, we leverage access to privileged information like prior works. We aim to improve exploration by crafting a reward bonus to recover the privileged information. However, the bonus should only incentivize gathering information that helps solve the task, and not other arbitrary information that the privileged information may hold. To achieve this, we extend the DREAM algorithm (Liu et al., 2021), which already designs such an exploration objective for the meta-RL setting. Specifically, we first apply an information bottleneck on the privileged information, which yields a necessary and sufficient representation of the privileged information for solving the task. Then, we add an exploration bonus derived from maximizing the mutual information between the explored observations and this representation of the privileged information. Importantly, this approach enables us to leverage flexible forms of privileged information beyond just traditional access to the unobserved state, such as transitions from future time-steps.

Overall, our main contribution is a new algorithm, which we call ***Pr****ivileged inf****o****rmation-****b****ased* ***e****xploration* (PROBE), that can learn targeted information-gathering strategies in POMDPs. PROBE contrasts existing POMDP algorithms that leverage privileged information, but are not designed to discover information-gathering strategies. Additionally, motivated by Humplik et al. (2019), we design PROBE to operate from arbitrary forms of privileged information, extending beyond just the full unobserved state. We also provide guarantees on the regret of our algorithm, and show that access to privileged information allows sample-efficient learning. In our experiments, PROBE successfully learns complex strategies in POMDPs with various exploration challenges, while remaining competitive on existing benchmark tasks that do not require sophisticated exploration. Further, PROBE can scale to POMDPs with egocentric pixel observations that require complex intra-episode exploration.

2. Related Work

The POMDP is a general framework for sequential decision-making given incomplete observations. There are no tractable exact methods as the agent's state space or action space grows (Kaelbling et al., 1998). However, there are approximate solutions that use recurrent neural networks to process histories of observations and actions (Hausknecht & Stone, 2015; Foerster et al., 2016; Zhu et al., 2017). To improve sample efficiency, more recent work leverages deep learning methods, such as those for representation learning (Kingma & Welling, 2013), to learn latent-space dynamics models of general POMDPs, including those with continuous state and action spaces (Watter et al., 2015; Wahlström et al., 2015; Karl et al., 2016; Igl et al., 2018; Han et al., 2019) and with high-dimensional image observations (Kapturowski et al., 2018; Hafner et al., 2019; Lee et al., 2020a).

Many works relax the POMDP by accessing privileged information at training time. Common to many algorithms is either an omniscient expert (Levine et al., 2016; Pan et al., 2017; Chen et al., 2020; Lee et al., 2020b; Warrington et al., 2021; Weihs et al., 2021; Kumar et al., 2021; Walsman et al., 2022; Shenfeld et al., 2023) or critic (Pinto et al., 2017; Baisero & Amato, 2021), or a learned dynamics model (Wang et al., 2023) that accesses latent state information at training time. These models facilitate the training of a separate policy, which operates without privileged information only from the observed state so that it can be deployed at test time. Theoretically, learning in tabular POMDPs with train-time latent state information is sample-efficient (Lee et al., 2023; Guo et al., 2023). However, while these existing methods use the privileged information to make solving the task easier, they either fail to address the issue of exploration (i.e., how to gather the information used by these models) or can only learn simple exploration strategies afforded by a general RL objective. Our work also leverages privileged information during training, but we explicitly address the issue of exploration to enable learning complex information-gathering policies.

There is a rich literature on such exploration in a special case of POMDPs studied in meta-RL (Wang et al., 2016; Humplik et al., 2019). The meta-RL setting typically considers a distribution over different tasks, represented by the latent part of the state, and the goal is to do well in a new task after only a few episodes of interaction. Many works therefore consider how to best leverage these few episodes to explore and gather information, known as the meta-exploration problem (Duan et al., 2016; Wang et al., 2016; Mishra et al., 2017; Gupta et al., 2018; Stadie et al., 2018; Zintgraf et al., 2019; Humplik et al., 2019; Kamienny et al., 2020; Liu et al., 2021; Zintgraf et al., 2021; Liu et al., 2022; Norman & Clune, 2023). Our work connects the meta-exploration problem with the general problem of exploring in POMDPs and proposes to generalize an existing meta-RL algorithm DREAM to learn from any form of privileged information in general POMDPs. Specifically, we leverage the mutual information objective from DREAM to learn to gather task-relevant information. everal prior works maximize mutual information objectives for RL exploration in fully-observable MDPs (Storck et al., 1995; Sun et al., 2011; Still & Precup, 2012; Houthooft et al., 2016), and for unsupervised skill learning (Gregor et al., 2016; Eysenbach et al., 2018; Warde-Farley et al., 2018). There has also been prior work that learns exploration policies to minimize error in the prediction of the learned dynamics model (Zhou et al., 2019) or the predicted privileged state (Margolis et al., 2023). Different from these objectives, our work learns to explore only with respect to task-relevant features, by applying an information bottleneck on the learned representation of the privileged state.

Leveraging hindsight information, e.g., by learning a future-conditioned value function, can accelerate reinforcement learning in an unknown environment (Harutyunyan et al., 2019; Schmidhuber, 2019; Guez et al., 2020; Mesnard et al., 2020; Venuto et al., 2021; Nota et al., 2021). Unlike these prior works, our algorithm incorporates the future trajectory through an exploration bonus. Many of the constraints introduced to reduce hindsight bias (Mesnard et al., 2020; Venuto et al., 2021) can also be readily combined with our proposed bonus.

3. Preliminaries and Problem Setup

Formally, a partially observable Markov decision process $\langle \mathcal{S}, \mathcal{A}, p, r, \mathcal{O}, f \rangle$ consists of a state space $\mathcal{S}$, action space $\mathcal{A}$, dynamics $p(s' \mid a, s)$, reward function $r(s, a)$, observation space $\mathcal{O}$ and observation function $f(o \mid s)$. The goal is to learn a policy $\pi(a_t \mid h_t)$ that produces actions $a_t \in \mathcal{A}$ conditioned on its history of observations, actions, and rewards $h_t = \{o_1, a_1, r_1, \ldots, o_t\}$ maximizing the expected cumulative rewards $\mathcal{J}(\pi) = \mathbb{E}_{\substack{s_t \sim p(\cdot \mid a_t, s_{t-1}) \\ a_t \sim \pi(\cdot \mid h_t)}} \left[\sum_{t=1}^{T} r(s_t, a_t) \right]$. Intuitively, the rewards and dynamics are controlled by the state s_t, but the agent does not directly observe the state and must instead make decisions based on its history h_t.

3.1. DREAM: Decoupling Exploration and Exploitation in Meta-Reinforcement Learning

Our work builds on the DREAM meta-RL algorithm (Liu et al., 2021). Specifically, we leverage the key insight that information-gathering exploration behavior can be learned from a weak form of privileged information in the meta-RL setting. DREAM achieves this by assuming access to a *task identifier* μ, which is a unique 1-hot identifier assigned to each different unobserved state seen during training, i.e., each different task has dynamics p^μ and r^μ. From a high level, DREAM consists of two stages: First, DREAM trains a policy conditioned on this task identifier with an information bottleneck, which yields a representation of the task identifier encoding the task-relevant information and a policy that can solve tasks given this task-relevant information. Second, since this task identifier representation contains all the information the policy needs to solve each task, DREAM learns to explore with an objective to obtain trajectories that recover the information inside the task identifier.

This idea is implemented in four main components:

1. An *encoder* $F_\psi(z \mid \mu)$ encodes the task identifier with an information bottleneck.
2. An *exploitation policy* $\pi_\theta^{\text{task}}(a \mid o, z)$ is trained conditioned on the task identifier encodings.
3. An *exploration policy* π_ϕ^{exp} maximizes the mutual information $I(\tau^{\text{exp}}; z)$ between its collected trajectories τ^{exp} and z sampled from the encoder.
4. A *decoder* $q_\omega(z \mid \tau^{\text{exp}})$ maps trajectories to encodings.

At test time, the exploration policy is first deployed to gather information about the task, and the exploitation policy conditions on the decoding of this trajectory to solve the task.

3.2. Learning with Privileged Information

We follow the formalism of the *informed POMDP* (Lambrechts et al., 2023) to leverage privileged information during training. Specifically, we augment the POMDP with an information space $\mathcal{I}$ and probability distribution $g(i \mid s)$. At each timestep t during training, the agent observes an information vector $i_t \sim g(\cdot | s_t)$. Our work studies different forms of privileged information vectors, including (i) the underlying state $i_t = s_t$; and (ii) future transitions $i_t = \tau_{t+1:T}$ seen later in the episode. At test time, the agent interacts with the original POMDP without privileged information.

4. Privileged Information-Based Exploration

We now introduce our approach called PROBE: ***Pr**ivileged **o**nformation-**b**ased **e**xploration*. Like prior works, we leverage privileged information to more easily learn the task. However, unlike prior work, our key idea is to additionally learn to explore and recover the privileged information. To do this, we first propose a reward bonus that maximizes the mutual information between the agent's observation and privileged information that is the hidden state (Section 4.1). Then, we extend this bonus to handle multiple forms of privileged information as well (Section 4.2). Finally, we provide a concrete implementation (Section 4.3).

4.1. An Information-Based Exploration Bonus

Naïvely, we could directly incentivize any exploration that recovers information about the hidden state s. However, the hidden state may contain information irrelevant to solving the task. For example, in our driving task, the state s may contain information about *all* roads, while only information about the roads leading to the destination matters. To discard task-irrelevant information, we learn an encoder $F_\psi(z \mid i)$ with an information bottleneck applied to z to eliminate excess information that the policy does not need. That is, we jointly train the encoder F_ψ and a policy π_θ, which conditions on z and the history h, with:

$$\max_{\psi,\theta} \mathbb{E}_{\substack{s_t \sim p(\cdot \mid a_t, s_{t-1}) \\ z_t \sim F_\psi(\cdot \mid i_t) \\ a_t \sim \pi_\theta(\cdot \mid h_t, z_t)}} \left[\sum_{t=1}^{T} r(s_t, a_t) - \lambda I(z_t; i_t) \right], \quad (1)$$

where $I(z_t; i_t)$ is the mutual information between z_t and i_t. Importantly, we will optimize both the policy π_θ and the encoder F_ψ to maximize the expected cumulative rewards. Alongside the information bottleneck, which only depends on the encoder, this objective encourages F_ψ to extract the task-relevant information and to discard everything else.

Critically, the privileged information i_t and, therefore, the embeddings z_t are available only during training. At test time, the agent no longer has access to this information. Hence, it is critical that, at test time, the agent's policy can collect the same information that z_t has, on its own. Put differently, we want the policy π_θ to develop exploration strategies that allow it to discover information about z_t. One way to do so is by maximizing $\sum_{t=1}^{T} I(h_T; z_t)$, the mutual information between the trajectory collected by the policy and the sequence of hidden states $z_{1:T}$. Following DREAM, we can efficiently optimize a variational lower bound of the mutual information (Barber & Agakov, 2004):

$$\begin{aligned} \sum_{t=1}^{T} I(h_T; z_t) &= \sum_{t=1}^{T} H(z_t) - \sum_{t=1}^{T} H(z_t \mid h_T) \\ &\geq \sum_{t=1}^{T} H(z_t) + \sum_{t=1}^{T} E_{z_t \sim F_\psi} \left[\log q_{\omega_t}(z_t \mid h_T) \right] \\ &= \sum_{t=1}^{T} H(z_t) + \sum_{t=1}^{T} \mathbb{E}_{z_t \sim F_\psi} \left[\log q_{\omega_t}(z_t \mid o_1) \right] \\ &\quad + \sum_{t=1}^{T} \mathbb{E}_{\substack{z_t \sim F_\psi, \\ h_T \sim \pi_\theta}} \left[\sum_{t'=1}^{T-1} \log \frac{q_\omega(z_t \mid h_{t'+1})}{q_\omega(z_t \mid h_{t'})} \right] \end{aligned}$$

where q is an arbitrary distribution with parameters ω. The last line expands a telescoping series. We can now break down this objective into a per-timestep reward, which can then be optimized by any RL algorithm. Since the first two terms do not depend on π_θ, we can ignore them when defining the reward bonus:

$$r_t^{\text{PROBE}} = \sum_{t'=1}^{T} \mathbb{E}_{z_t \sim F_\psi} \left[\log \frac{q_\omega(z_{t'} \mid h_{t+1})}{q_\omega(z_{t'} \mid h_t)} \right]. \quad (2)$$

This reward measures the additional information gained about the full sequence of hidden states $z_1, z_2, \ldots, z_T$ from the transition to the observed o_{t+1} and r_t. While there are many ways to combine the task and exploration rewards, we will take their sum, scaled by a factor α: $\hat{r}_t = r_t + \alpha r_t^{\text{PROBE}}$.

Algorithm 1 PROBE (single train episode)

```
Input: use privileged info
$h_1 = \{o_1\}$
for $t = 1, 2, \ldots, T$ do
  if use privileged info then
    Compute encoding of privileged info $z_t \sim F_\psi(\cdot \mid i_t)$
  else
    Sample $z_t \sim q_\omega(\cdot \mid h_t)$
  end if
  Roll out policy $a_t \sim \pi_\theta(\cdot \mid h_t, z_t)$
  Update history $h_{t+1} = h_t \cup \{a_t, o_{t+1}, r_t\}$
end for
Relabel rewards $\hat{r}_t = r_t + \alpha r_t^{\text{PROBE}}$ (defined in Eqn. 3)
Update $\pi_\theta$ and $F_\psi$ to maximize Eqn. 1 with rewards $\hat{r}_t$
Update $q_\omega$ to minimize $\sum_{t=1}^{T} \mathbb{E}_{z \sim F_\psi(i_t)} \left[ \|z - q_\omega(h)\|_2^2 \right]$
```

4.2. General Forms of Privileged Information

In the previous section, we defined the encoder F_ψ to represent the hidden state s_t, but in principle, this encoder can represent *any* privileged information (PI) available during training. The flexibility to leverage different forms of PI is particularly powerful in situations where the ground-truth hidden states cannot be accessed even during training. In such cases, we can look to alternative, less presumptive sources of information, such as expert demonstrations, or even information that we can obtain at no additional cost, such as transitions from future time-steps. Furthermore, the information bottleneck that we apply to z_t means that even if the PI contains task-irrelevant features, they will removed from the representation. Therefore, the PROBE rewards (1) will be robust to choices of PI that contain excess features, and (2) can benefit from choices that contain more information about the hidden states.

One particularly interesting form of information is the remainder of a trajectory, $i_t = \tau_{t+1:T}$. The full trajectory is privileged in the sense that it is only available during training, after we have rolled out the policy for the full episode. However, this information comes at no additional cost, unlike hidden states or expert actions. Intuitively, the future transitions from the same trajectory can sometimes contain information about the current and past hidden states. For instance, when driving down a road with traffic, we may not discover whether the road is blocked until we have already traveled a substantial part of the road. However, given this full trajectory, we can *in hindsight* encourage exploration strategies, e.g., reading nearby signs, that indicate whether the road is blocked before even navigating down the road.

4.3. Practical Implementation

The components of PROBE are the privileged information encoder $F_\psi(z \mid i)$, decoder $q_\omega(z \mid h)$, and policy $\pi_\theta(a \mid h, z)$.

The privileged information encoder $F_\psi(z \mid i)$ consists of a deterministic encoder $f_\psi(i)$ with Gaussian noise applied. By setting the prior to be a Gaussian with the same variance, the information bottleneck then becomes an ℓ_2 regularization on $f_\psi(i)$. The decoder $q_\omega(z \mid h)$ is similarly parameterized by a deterministic encoder $g_\omega(h)$, with Gaussian noise of the same variance added. Therefore, maximizing $\mathbb{E}_{z\sim F_\psi(i)}\left[\log q_\omega(z \mid h)\right]$ is equivalent to minimizing $\mathbb{E}_{z\sim F_\psi(i)}\left[\|z - g_\omega(h)\|_2^2\right]$, and the exploration reward is equivalent to:

$$r_t^{\text{PROBE}} = \sum_{t'=1}^{T} \mathbb{E}_{z_{t'}\sim F_\psi}\left[\log q_\omega(z_{t'} \mid h_{t+1}) - \log q_\omega(z_{t'} \mid h_t)\right]$$

$$= \sum_{t'=1}^{T} \|f_\psi(i_{t'}) - g_\omega(h_t)\|_2^2 - \|f_\psi(i_{t'}) - g_\omega(h_{t+1})\|_2^2 .$$

Temporal locality. This reward is currently a summation of the information gain for all hidden states of the episode. To ensure that the mutual information term is correctly estimated, we would need a separate decoder for each timestep, i.e., $\omega = (\omega_t)_{t=1}^T$. Instead, we will simplify the reward bonus with temporal locality. Specifically, we will design the reward so that at time-step t, the bonus will only be derived from the information gained about z_{t+1}. This is a reasonable simplification, because (1) conditioned on z_{t+1}, the past states $z_{1:t}$ will no longer matter for future decision-making and (2) it is generally unlikely for the agent to learn much about states in the distant future. The simplified bonus can be expressed as

$$r_t^{\text{PROBE}} = \|f_\psi(i_{t+1}) - g_\omega(h_t)\|_2^2 - \|f_\psi(i_{t+1}) - g_\omega(h_{t+1})\|_2^2 . \tag{3}$$

Finally, the policy $\pi_\theta(a \mid h, z)$ takes in an encoding z, which at test time, is sampled from the history encoder $q_\omega(z \mid h)$. During training, we sample z from $F_\psi(z \mid i)$ during every other episode to improve training efficiency. We summarize the PROBE algorithm in Algorithm 1 and illustrate all described components in Fig. 1.

When using the future transitions, we parameterize f_ψ as a recurrent neural network to encode the full trajectory h_T, and the embedding serves as the privileged information at each timestep. We refer this version of our algorithm as PROBE-FUTURE. An important hyperparameter that PROBE introduces on top of DREAM is α, the scaling factor for the exploration rewards. This factor controls the weight of the exploration bonus relative to the task rewards. In principle, the multiplier α can be annealed to zero over training so that the final objective is unbiased, but in practice, we find that a carefully selected constant is sufficient. See App. A for more details about the implementation of PROBE.

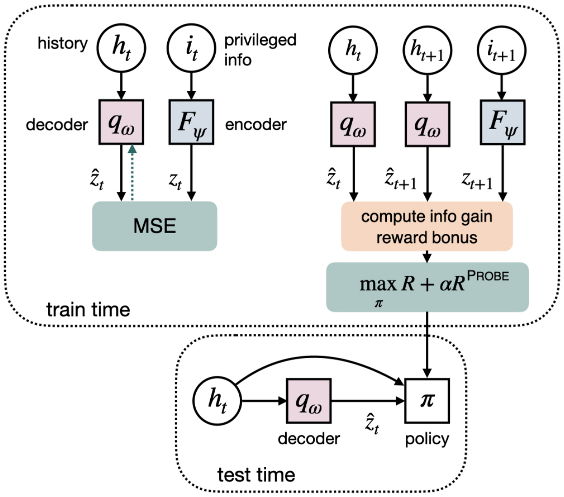

Figure 1. PROBE learns an encoder F_ψ which encodes the privileged information i_t to z_t, and a decoder q_ω which maps the history h_t to $\hat{z}_t$ and is trained to minimize the loss to z_t. Finally, it constructs the exploration bonus via Eqn. 3, and trains a policy π_θ to maximize the sum of the task and exploration rewards.

5. Theoretical Analysis of PROBE

In contrast to MDPs, finding a near-optimal policy in POMDPs requires a number of samples at least exponential in the horizon length T in the worst-case (Krishnamurthy et al., 2016). This sample-inefficiency is largely attributed to the fact that the agent may not observe any useful information about the true underlying state of the system, reducing the problem to something like a blind tree search in the worst case. However, we will show theoretically that access to privileged information during training allows us to circumvent this worst case and achieve sample-efficient learning, providing a formal motivation for PROBE.

Notation. We consider the *Bayesian regret* incurred over L episodes of interactions by a policies iteratively learned via PROBE. As notation, denote by $\widetilde{h}_{\ell,t} = (i_1^\ell, a_1^\ell, r_1^\ell, \ldots, i_t^\ell, a_1^\ell, r_t^\ell)$ the history of episode $\ell \in [L]$ until timestep $t \in [T]$ *using the privileged information*, which in our setting is additionally observed during train time.

Let $\widetilde{\mathcal{D}}_\ell = (\widetilde{h}_{1,T}, \ldots, \widetilde{h}_{\ell-1,T})$ be the dataset of histories, and let $\mathcal{E}$ consist of the parameters of the POMDP, specifically the transition and reward functions. The cumulative regret of an algorithm $\Pi = \{\pi^\ell\}_{\ell=1}^L$ that deploys policy π^ℓ during episode L in environment $\mathcal{E}$ is defined as $\mathcal{R}(L, \Pi, \mathcal{E}) = \sum_{\ell=1}^L \mathcal{J}(\pi^*, \mathcal{E}) - \mathcal{J}(\pi^\ell, \mathcal{E})$. Finally, the Bayesian regret takes an expectation over the prior distribution over environment $\mathcal{E}$, such that $\mathcal{BR}(L, \Pi) = \mathbb{E}[\mathcal{R}(L, \Pi, \mathcal{E})]$.

Algorithm. We consider an oracle version of the PROBE algorithm, where at episode L, we deploy a policy π^ℓ conditioned on history that satisfies:

$$\pi^\ell = \arg\max_\pi \mathbb{E}\left[\sum_{t=1}^{T} r(s_t, a_t) + \alpha I(\widetilde{h}_{t-1}; i_t) \mid \widetilde{\mathcal{D}}_\ell\right] \quad (4)$$

where the expectation is taken over environment $\mathcal{E}$ and policy π. Recall that in practice, PROBE instead utilizes a variational approximation of $I(h_{t-1}; z_t)$ as the reward bonus, where z_t is a learned embedding of information i_t, because we do not have access to i_t during evaluation.

Regret bound. We show that our analyzed version of PROBE enjoys polynomial regret under the following assumption about the provided privileged information:

Assumption 5.1. The privileged information satisfies being a deterministic function of hidden state such that: for any timestep $t \in [T]$, $p(\cdot \mid s_t, a_t) = p(\cdot \mid i_t, a_t)$ for $i_t = g(s_t)$.

This assumption means that the privileged information, in conjunction with the observed history, is sufficient to predict the next hidden state. Note that this is trivially true for $i = s$. Using Assumption 5.1, we can prove the following Bayesian regret bound for our algorithm.

For simplicity, we also assume that the reward function $r_t = r(s_t, a_t)$ is deterministic. This assumption is only for ease of algebra, and our derivation can be easily adapted to include stochastic reward and observations.

Theorem 5.2. *In a tabular POMDP, by choosing* $\alpha = \widetilde{\mathcal{O}}(\sqrt{LT^2/|\mathcal{S}|})$ *in Equation* (4)*, the Bayesian regret of our algorithm can be bounded as:*

$$\mathcal{BR}(L, \Pi) \leq \sqrt{8LT^4|\mathcal{S}|^2|\mathcal{A}|^2|\mathcal{I}|\log(LT|\mathcal{S}|)}\,.$$

We defer a full proof of Theorem 5.2 to Appendix D. Our derivation leverages recent results in information-directed sampling (Russo & Roy, 2014; Hao & Lattimore, 2022); however, the algorithms considered in such work are for MDPs and often intractable to implement in practice. In contrast, PROBE is both practical and achieves comparable regret bounds for POMDPs with privileged information.

6. Experiments

Our experiments aim to study whether PROBE can learn effective exploration strategies across various POMDP problems with privileged state information at training time. We also evaluate whether PROBE can learn from future transitions, while remaining robust to suboptimal choices of privileged information, such as task-irrelevant information. Videos and code can be found at https://sites.google.com/view/probe-explore-icml.

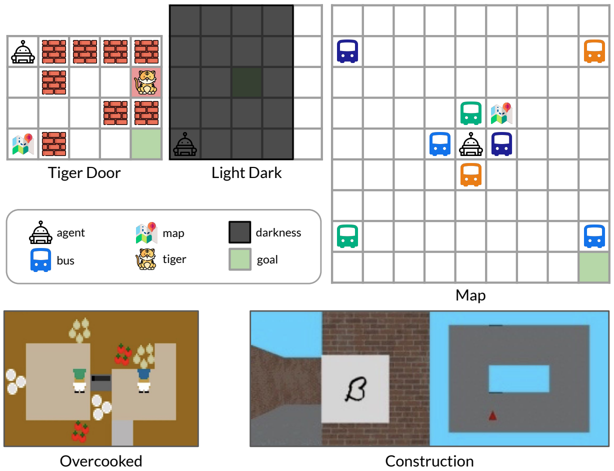

Figure 2. Environment visualizations. The *Construction* environment is depicted in the top-down view on the right for clarity, but the agent only receives the first-person view on the left as state.

6.1. Experimental Setup

Environments. We begin our experiments with simple partially observable problems studied by prior work. All environments are visualized in Fig. 2.

- **Tiger Door** (Littman et al., 1995). This gridworld has a goal and a trap cell at the end of two hallways. The positions of these cells are unknown but revealed by visiting the map cell. The goal cell and trap cell give a reward of $+1$ and -1 respectively, and terminate the episode.
- **Light-Dark** (Platt Jr et al., 2010). The agent is randomly initialized in the "dark" part of a room, where the agent receives an uninformative observation of its position (a constant of $(0, 0)$). Its goal is to reach the goal at the center. To observe its true position, the agent can visit the "light" part, along the right side of the room.
- **Map** (Liu et al., 2021). The agent can reach the goal cell more quickly by taking one of the four buses. The configuration of bus source-destinations is unknown but revealed by visiting a map cell.

We further introduce three new POMDP problems that require more complex exploration strategies.

- **Non-stationary Map**. A non-stationary version of Map, in which the bus routes change every 2 to 3 time-steps. It requires the agent 2 steps to take any of the buses from the map state, therefore the agent will only successfully reach its destination if the routes change after 3 steps.
- **Overcooked** (Carroll et al., 2019). The agent needs to prepare two dishes with a second agent by placing the correct ingredient in the pot and serving the finished dishes. The correct ingredient is determined by the other ingredients that the second agent has filled the pot with. The recipe that the other agent is following is only known by visiting the corner of the kitchen and reading the order.

- **Construction.** This vision-based environment, built on top of Miniworld (Chevalier-Boisvert et al., 2023), has two routes that take the agent to its goal destination. Each route is either clear or blocked due to construction, which is indicated by a sign near the start of the route and by construction cones that are only in view at the end of the route. The status of a route, i.e., whether it is clear or blocked, can change during an episode.

Of these domains, Non-stationary Map is designed to test the ability to learn exploration strategies that *generalize* to new conditions not seen during training. See App. B for more details about these environments, such as their observation and action spaces.

Comparisons. For the first of our comparisons, we evaluate a recurrent policy that takes an input the history and outputs an action, trained via double deep Q-learning (Van Hasselt et al., 2016). We also consider multiple baselines designed for POMDPs and use some form of privileged information.

- **IMPORT** (Kamienny et al., 2020): POMDP algorithm that, like PROBE, trains an encoder $F_\psi(z \mid s)$, decoder $q_\omega(z \mid h)$, and policy $\pi(a \mid h, z)$. IMPORT uses an auxiliary loss function between the history embedding from the decoder and the embedding from the encoder, which is optimized only with respect to the parameters of the decoder.
- **ELF** (Walsman et al., 2022): POMDP algorithm that uses action labels from an omniscient expert, which acts optimally with respect to the underlying state. It trains a follower policy, which is not privileged, to imitate the omniscient expert, and uses the estimated value of this policy to shape the rewards of an explorer policy.
- **DREAM** (Liu et al., 2021): Meta-RL algorithm that decouples the exploration policy from the exploitation policy. To adapt it to our setting, which does not have a separate exploration episode, we augment the exploration policy with an "end episode" action that will switch from the exploration to the exploitation policy. Unlike PROBE, this comparison performs one stage of exploration.

See App. A for more details about these methods.

6.2. Learning in Benchmark Environments

We first verify that it can learn strong policies in commonly-studied POMDP problems. In Fig. 3 (top), we compare the average performance of PROBE to prior methods after training on 100K episodes, and find that PROBE achieves similar or better performance than prior POMDP and meta-RL algorithms. Because PROBE is able to learn a task-relevant exploration strategy, the agent recovers the highest achievable return in all tasks.

All comparisons, with the exception of the recurrent policy, have access to some form of privileged information during training. The DREAM and IMPORT agents have the task identifier, while ELF has expert actions. However, DREAM is the only comparison that also recovers the optimal return across all three domains. We observe that the recurrent agent eventually learns the optimal exploration behavior but less efficiently than PROBE and DREAM. In contrast, the IMPORT and ELF agents do not learn to gather information and instead will directly try to solve the task without it. However, in Tiger Door and Light Dark, this information is critical to solving the task. Therefore, both IMPORT and ELF agents, which make a random guess about the task, will only be correct a fraction of the time. In Map, reading the map and taking the correct bus is the optimal solution, but the agent can still reach the goal by directly walking to it, which takes longer. Both IMPORT and ELF agents learn the latter behavior. Because ELF accesses the expert actions, we also find that it learns quickly initially but fails to explore reliably to identify the task.

6.3. More Complex Exploration Strategies

We now evaluate our method on the new POMDP problems described in Section 6.1. In contrast to the problems in the previous experiments, these necessitate multiple rounds of information gathering, due to the dynamism of the task. In Fig. 3 (bottom), we present the results from these domains. In Nonstationary Map, PROBE exhibits the desired behavior of alternating between reading the map and taking the bus according to the present state of the environment, until it successfully reaches its target destination. In Overcooked, the agent correctly reads the placed order each time before starting to place ingredients into the pot.

Due to its single exploration phase, we observe that the DREAM exploration agent will either (1) switch control over to the exploitation agent after exploring once or (2) never stop exploration. With the former behavior, the exploitation agent fails to produce the same exploration behavior once the environment changes. With the latter, the agent demonstrates the desired behavior of re-exploring as changes occur, but never makes progress towards completing the task because it is not trained with task rewards, like PROBE. In Overcooked, the other comparisons have similar failure modes to those described in the benchmark problems. In Non-stationary Map, IMPORT and ELF achieve final returns of 0.33 and 0.29, which are close to PROBE's return of 0.41, but they achieve this by *walking* to the goal rather than taking the buses, which on average is more efficient.

Scaling to image observations. Construction is a challenging 3D visual navigation task with sparse rewards, which we use to evaluate the scalability of PROBE to high-dimensional observations. In this domain, the agent must decide which road to take by reading signs that indicate whether the road is blocked due to construction. Because the status of the road can change at any (unknown) point, the agent may be

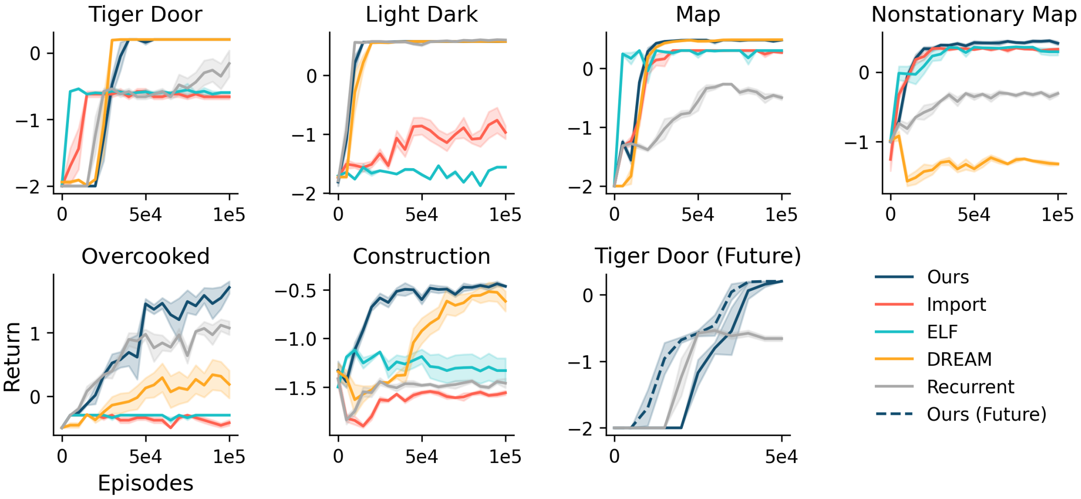

Figure 3. Performance on all environments. PROBE and DREAM achieve optimal performance across all three benchmark domains. Second row, third column: Using future transitions as the privileged information (PROBE (FUTURE)) results in similar performance as PROBE. Our exploration bonus can be used even without privileged information. Error bars represent the standard error over 5 seeds.

required to change its route even after checking the sign.

The PROBE agent learns to read the signs, and even waits at the beginning of the road if it is clear for a few extra time-steps. The purpose of this behavior is to avoid walking down the road too early in case the status of the road changes soon after. If the status does change, the agent saves the few time-steps that it would have taken to walk down the road and turn around. On the other hand, the DREAM agent does not learn this hedging behavior. Its exploration policy reads the sign and immediately switches to the exploitation policy. The second policy will then walk down the road, and if the status changes, it will see the construction cones at the end of the road after which it will turn around. We believe the hedging behavior shown by PROBE is possible due to the joint optimization of the exploration and task rewards. The recurrent, ELF, and IMPORT agents do not reliably read the signs and rely on the cones that appear at the end of the road to decide whether to change routes.

6.4. Learning from Future Transitions

As discussed in Section 4.2, our method can in principle learn from alternative forms of information. In this experiment, we explore the use of future transitions from an episode: $i_t = \tau_{t+1:T}$, and we call this variant of our algorithm PROBE (FUTURE). Because future transitions are not truly privileged in that they are available to all online algorithms, this variant can be applied to any decision-making problem, regardless of whether it offers additional information. Fig. 3 (second row, third column) compares the performance of PROBE and PROBE (FUTURE) in the Tiger Door domain. We see that the variant without privileged information performs similarly to PROBE and even slightly improves the learning efficiency. These results suggest that PROBE (FUTURE) is a competitive algorithm that can be applied to any POMDP problem with *any* additional assumptions, and achieve results similar to algorithms that use explicit privileged information.

6.5. Sensitivity Analysis

Next, we evaluate the robustness of our algorithm to different hyperparameters and types of privileged information. Below, we study the choice of the weight on the PROBE exploration bonus. In App. C, we study the sensitivity to excess and noisy privileged information.

Weight on exploration bonus. The choice of α, which determines the importance of the exploration bonus in the overall objective, can impact the performance of the resulting PROBE policy. In this experiment, we evaluate PROBE with $\alpha \in \{0.0, 0.1, 0.5, 1.0, 2.0\}$ in the Tiger Door domain, and report the results in Fig. 4. We find that the choice of α can significantly impact the final performance of the PROBE policy. When the multiplier is too small, the scaled reward bonus may not be enough to incentivize exploration. However, when it is too large, it may outweigh the task rewards and lead to unsuccessful policies. Therefore, we recommend choosing the largest possible α value such that the total scaled PROBE reward does not exceed the optimal task reward, i.e., $\sum_t \alpha r_t^{\text{PROBE}} < \sum_t r_t$. We find these policies to be the most successful in our experiments, as this

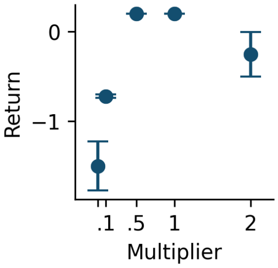

Figure 4. PROBE with different weights on the exploration bonus.

ensures that the exploration and task rewards are weighted roughly equally.

7. Discussion

Many partially-observable decision-making problems require complex information-gathering strategies to successfully complete the task. To approach these settings, we designed an exploration bonus that rewards information gathered about the hidden state, or more generally about any privileged information available during training. In contrast, prior algorithms leveraging similar privileged information either only work in stationary meta-RL settings or fail to recover the optimal exploration strategy. In experiments on multiple partially-observed environments, our exploration bonus leads to more efficient exploration strategies and as a result, improved task rewards. We also proposed a more general variant of our algorithm that does not need any privileged information, and observed its efficacy.

Limitations and future work. There are limitations to our work, which only studies partially-observable settings that require the agent to gather new information whenever the hidden component of the state changes. In some scenarios, the state may shift in a structured way that can be predicted with high accuracy, and exploration is unnecessary in such cases. In principle, PROBE can handle this type of non-stationarity by modifying the design of the decoder. Our experiments also focused on domains with finite hidden state spaces, so extending PROBE to more diverse environments with larger or continuous state spaces would be an exciting avenue to explore in future work.

Impact Statement

Our algorithm requires access to privileged information specified by the algorithm designer. This means that, if misspecified, certain features may be overlooked while other features may be overemphasized. While we designed an information-bottlenecked representation to prevent overemphasis of task-irrelevant features, missing features may result in policies with undesirable behaviors. An example of this is a recommendation system that interacts with diverse users. Failure to include critical features, e.g., demographic information of the users, in the design of the privileged information can lead to potentially negative or harmful user interactions.

References

Baisero, A. and Amato, C. Unbiased asymmetric actor-critic for partially observable reinforcement learning. *arXiv preprint arXiv:2105.11674*, 2021.

Barber, D. and Agakov, F. The im algorithm: a variational approach to information maximization. *Advances in neural information processing systems*, 16(320):201, 2004.

Carroll, M., Shah, R., Ho, M. K., Griffiths, T., Seshia, S., Abbeel, P., and Dragan, A. On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems*, 32, 2019.

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

Chevalier-Boisvert, M., Dai, B., Towers, M., de Lazcano, R., Willems, L., Lahlou, S., Pal, S., Castro, P. S., and Terry, J. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *arXiv preprint arXiv:2306.13831*, 2023.

Doshi-Velez, F. and Konidaris, G. Hidden parameter markov decision processes: A semiparametric regression approach for discovering latent task parametrizations. In *IJCAI: proceedings of the conference*, volume 2016, pp. 1432. NIH Public Access, 2016.

Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., and Abbeel, P. Rl2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

Foerster, J., Assael, I. A., De Freitas, N., and Whiteson, S. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

Gregor, K., Rezende, D. J., and Wierstra, D. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

Guez, A., Viola, F., Weber, T., Buesing, L., Kapturowski, S., Precup, D., Silver, D., and Heess, N. Value-driven hindsight modelling. *Advances in Neural Information Processing Systems*, 33:12499–12509, 2020.

Guo, J., Chen, M., Wang, H., Xiong, C., Wang, M., and Bai, Y. Sample-efficient learning of pomdps with multiple observations in hindsight. *arXiv preprint arXiv:2307.02884*, 2023.

Gupta, A., Mendonca, R., Liu, Y., Abbeel, P., and Levine, S. Meta-reinforcement learning of structured exploration strategies. *Advances in neural information processing systems*, 31, 2018.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019.

Han, D., Doya, K., and Tani, J. Variational recurrent models for solving partially observable control tasks. *arXiv preprint arXiv:1912.10703*, 2019.

Hao, B. and Lattimore, T. Regret bounds for information-directed reinforcement learning. In *Advances in neural information processing systems*, 2022.

Harutyunyan, A., Dabney, W., Mesnard, T., Gheshlaghi Azar, M., Piot, B., Heess, N., van Hasselt, H. P., Wayne, G., Singh, S., Precup, D., et al. Hindsight credit assignment. *Advances in neural information processing systems*, 32, 2019.

Hausknecht, M. and Stone, P. Deep recurrent q-learning for partially observable mdps. In *2015 aaai fall symposium series*, 2015.

Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., and Abbeel, P. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

Humplik, J., Galashov, A., Hasenclever, L., Ortega, P. A., Teh, Y. W., and Heess, N. Meta reinforcement learning as task inference. *arXiv preprint arXiv:1905.06424*, 2019.

Igl, M., Zintgraf, L., Le, T. A., Wood, F., and Whiteson, S. Deep variational reinforcement learning for pomdps. In *International Conference on Machine Learning*, pp. 2117–2126. PMLR, 2018.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Kamienny, P.-A., Pirotta, M., Lazaric, A., Lavril, T., Usunier, N., and Denoyer, L. Learning adaptive exploration strategies in dynamic environments through informed policy regularization. *arXiv preprint arXiv:2005.02934*, 2020.

Kapturowski, S., Ostrovski, G., Quan, J., Munos, R., and Dabney, W. Recurrent experience replay in distributed reinforcement learning. In *International conference on learning representations*, 2018.

Karl, M., Soelch, M., Bayer, J., and Van der Smagt, P. Deep variational bayes filters: Unsupervised learning of state space models from raw data. *arXiv preprint arXiv:1605.06432*, 2016.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Krishnamurthy, A., Agarwal, A., and Langford, J. Contextual-mdps for pac-reinforcement learning with rich observations. In *Advances in neural information processing systems*, volume 29, 2016.

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

Lambrechts, G., Bolland, A., and Ernst, D. Informed pomdp: Leveraging additional information in model-based rl. *arXiv preprint arXiv:2306.11488*, 2023.

Lee, A. X., Nagabandi, A., Abbeel, P., and Levine, S. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *Advances in Neural Information Processing Systems*, 33:741–752, 2020a.

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

Lee, J., Agarwal, A., Dann, C., and Zhang, T. Learning in pomdps is sample-efficient with hindsight observability. In *International Conference on Machine Learning*, pp. 18733–18773. PMLR, 2023.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Littman, M. L., Cassandra, A. R., and Kaelbling, L. P. Learning policies for partially observable environments: Scaling up. In *Machine Learning Proceedings 1995*, pp. 362–370. Elsevier, 1995.

Liu, E., Stephan, M., Nie, A., Piech, C., Brunskill, E., and Finn, C. Giving feedback on interactive student programs with meta-exploration. *Advances in Neural Information Processing Systems*, 35:36282–36294, 2022.

Liu, E. Z., Raghunathan, A., Liang, P., and Finn, C. Decoupling exploration and exploitation for meta-reinforcement learning without sacrifices. In *International conference on machine learning*, pp. 6925–6935. PMLR, 2021.

Margolis, G. B., Fu, X., Ji, Y., and Agrawal, P. Learning to see physical properties with active sensing motor policies. *arXiv preprint arXiv:2311.01405*, 2023.

Mesnard, T., Weber, T., Viola, F., Thakoor, S., Saade, A., Harutyunyan, A., Dabney, W., Stepleton, T., Heess, N., Guez, A., et al. Counterfactual credit assignment in model-free reinforcement learning. *arXiv preprint arXiv:2011.09464*, 2020.

Mishra, N., Rohaninejad, M., Chen, X., and Abbeel, P. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

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

Norman, B. and Clune, J. First-explore, then exploit: Meta-learning intelligent exploration. *arXiv preprint arXiv:2307.02276*, 2023.

Nota, C., Thomas, P., and Da Silva, B. C. Posterior value functions: Hindsight baselines for policy gradient methods. In *International Conference on Machine Learning*, pp. 8238–8247. PMLR, 2021.

Pan, Y., Cheng, C.-A., Saigol, K., Lee, K., Yan, X., Theodorou, E., and Boots, B. Agile autonomous driving using end-to-end deep imitation learning. *arXiv preprint arXiv:1709.07174*, 2017.

Pinto, L., Andrychowicz, M., Welinder, P., Zaremba, W., and Abbeel, P. Asymmetric actor critic for image-based robot learning. *arXiv preprint arXiv:1710.06542*, 2017.

Platt Jr, R., Tedrake, R., Kaelbling, L., and Lozano-Perez, T. Belief space planning assuming maximum likelihood observations. 2010.

Russo, D. and Roy, B. V. Learning to optimize via information directed sampling. *CoRR*, abs/1403.5556, 2014.

Schmidhuber, J. Reinforcement learning upside down: Don't predict rewards–just map them to actions. *arXiv preprint arXiv:1912.02875*, 2019.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shenfeld, I., Hong, Z.-W., Tamar, A., and Agrawal, P. Tgrl: Teacher guided reinforcement learning algorithm for pomdps. In *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023.

Stadie, B. C., Yang, G., Houthooft, R., Chen, X., Duan, Y., Wu, Y., Abbeel, P., and Sutskever, I. Some considerations on learning to explore via meta-reinforcement learning. *arXiv preprint arXiv:1803.01118*, 2018.

Still, S. and Precup, D. An information-theoretic approach to curiosity-driven reinforcement learning. *Theory in Biosciences*, 131:139–148, 2012.

Storck, J., Hochreiter, S., Schmidhuber, J., et al. Reinforcement driven information acquisition in non-deterministic environments. In *Proceedings of the international conference on artificial neural networks, Paris*, volume 2, pp. 159–164, 1995.

Sun, Y., Gomez, F., and Schmidhuber, J. Planning to be surprised: Optimal bayesian exploration in dynamic environments. In *Artificial General Intelligence: 4th International Conference, AGI 2011, Mountain View, CA, USA, August 3-6, 2011. Proceedings 4*, pp. 41–51. Springer, 2011.

Van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Venuto, D., Lau, E., Precup, D., and Nachum, O. Policy gradients incorporating the future. *arXiv preprint arXiv:2108.02096*, 2021.

Wahlström, N., Schön, T. B., and Deisenroth, M. P. From pixels to torques: Policy learning with deep dynamical models. *arXiv preprint arXiv:1502.02251*, 2015.

Walsman, A., Zhang, M., Choudhury, S., Fox, D., and Farhadi, A. Impossibly good experts and how to follow them. In *The Eleventh International Conference on Learning Representations*, 2022.

Wang, A., Li, A. C., Klassen, T. Q., Icarte, R. T., and McIlraith, S. A. Learning belief representations for partially observable deep rl. 2023.

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

Warde-Farley, D., Van de Wiele, T., Kulkarni, T., Ionescu, C., Hansen, S., and Mnih, V. Unsupervised control through non-parametric discriminative rewards. *arXiv preprint arXiv:1811.11359*, 2018.

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

Watter, M., Springenberg, J., Boedecker, J., and Riedmiller, M. Embed to control: A locally linear latent dynamics model for control from raw images. *Advances in neural information processing systems*, 28, 2015.

Weihs, L., Jain, U., Liu, I.-J., Salvador, J., Lazebnik, S., Kembhavi, A., and Schwing, A. Bridging the imitation gap by adaptive insubordination. *Advances in Neural Information Processing Systems*, 34:19134–19146, 2021.

Zhou, W., Pinto, L., and Gupta, A. Environment probing interaction policies. *arXiv preprint arXiv:1907.11740*, 2019.

Zhu, P., Li, X., Poupart, P., and Miao, G. On improving deep reinforcement learning for pomdps. *arXiv preprint arXiv:1704.07978*, 2017.

Zintgraf, L., Shiarlis, K., Igl, M., Schulze, S., Gal, Y., Hofmann, K., and Whiteson, S. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. *arXiv preprint arXiv:1910.08348*, 2019.

Zintgraf, L. M., Feng, L., Lu, C., Igl, M., Hartikainen, K., Hofmann, K., and Whiteson, S. Exploration in approximate hyper-state space for meta reinforcement learning. In *International Conference on Machine Learning*, pp. 12991–13001. PMLR, 2021.

A. Implementation Details

A.1. PROBE (Ours)

The PROBE policy π_θ is parameterized as a recurrent deep dueling double-Q network (Van Hasselt et al., 2016). To train this policy along with the encoder F_ψ and decoder q_ω, we sample from the replay buffer a tuple of $(h_t, i_t, a_t, r_t, o_{t+1}, i_{t+1})$ and define the following losses:

$$\begin{aligned}
\mathcal{L}(\psi) &= \mathbb{E}\left[\min(\|f_\psi(i)\|_2^2, K)\right] \\
\mathcal{L}(\omega) &= \mathbb{E}\left[\|f_\psi(i) - g_\omega(h)\|_2^2\right] \\
\mathcal{L}(\theta, \psi) &= \mathbb{E}\left[\|\hat{Q}_\theta(h, f_\psi(i), a) - (\hat{r} + \hat{Q}_{\theta'}(h', f_\psi(i'), a'))\|_2^2\right], \\
&\quad \text{where } a' = \arg\max_a \hat{Q}_\theta(h', f_\psi(i'), a), \hat{r} = r + \alpha r^{\text{PROBE}} \\
\mathcal{L}(\theta, \omega) &= \mathbb{E}\left[\|\hat{Q}_\theta(h, g_\omega(h), a) - (\hat{r} + \hat{Q}_{\theta'}(h', g_\omega(h'), a'))\|_2^2\right] \\
&\quad \text{where } a' = \arg\max_a \hat{Q}_\theta(h', g_\omega(h'), a), \hat{r} = r + \alpha r^{\text{PROBE}}
\end{aligned}$$

and where K is a hyperparameter and θ' are the parameters of the target network. For all of our experiments, we choose $K = 10$ following DREAM. We minimize the sum of these four losses, and periodically update the target network.

The privileged information and observations are embedded in the same way, which is described next. Each dimension of the privileged information/observation is embedded with an embedding matrix to an output of dimension 32 and concatenated together. They are then passed through a 2-layer MLP with dimensions 256 and 64. The history h is broken up into tuples of (a_t, r_t, o_{t+1}, d_t), where d_t represents whether the episode is done at timestep t. The action a_t, reward r_t, and done flag d_t are separately embedded to an output of dimension of 16 each. The observation o_{t+1} is embedded as previously described; then all components are concatenated and passed through a linear layer with output dimension 64. Each tuple is embedded following the above scheme and passed into an LSTM with hidden dimension 64 to obtain the embedding of the overall history. In Construction, the observations are images and are embedded by a CNN with 3 layers, each with 32 features and stride of 2, and with filter sizes of 5, 5, and 4. The output of the CNN is 128-dimensional.

Clipped distances. The PROBE reward is defined in Eqn. 3 and reproduced below:

$$r_t^{\text{PROBE}} = \|f_\psi(i_{t+1}) - g_\omega(h_t)\|_2^2 - \|f_\psi(i_{t+1}) - g_\omega(h_{t+1})\|_2^2.$$

We clip this bonus before adding it to the task reward, so that we can better set the weight α. Specifically, we clip each distance term with D:

$$\begin{aligned}
r_t^{\text{PROBE,clipped}} &= \min\left(\|f_\psi(i_{t+1}) - g_\omega(h_t)\|_2^2, D\right) \\
&\quad - \min\left(\|f_\psi(i_{t+1}) - g_\omega(h_{t+1})\|_2^2, D\right),
\end{aligned}$$

where we choose $D = 1.0$ for all of our experiments.

A.2. Recurrent Policy

The recurrent policy π_θ is also parameterized as a DDQN, which is optimized with the following loss:

$$\begin{aligned}
\mathcal{L}(\theta) &= \mathbb{E}\left[\|\hat{Q}_\theta(h, a) - (r + \hat{Q}_{\theta'}(h', a'))\|_2^2\right] \\
&\text{where } a' = \arg\max_a \hat{Q}_\theta(h', a).
\end{aligned}$$

Similar to PROBE, θ' are the parameters of the target network, which are periodically updated. The embedding scheme is the same as that of PROBE.

A.3. IMPORT

The IMPORT policy π_θ is also parameterized as a recurrent DDQN. Like PROBE, there is an encoder F_ψ and decoder q_ω. The losses are:

$$\begin{aligned}
\mathcal{L}(\omega) &= \mathbb{E}\left[\|f_\psi(i) - g_\omega(h)\|_2^2\right] \\
\mathcal{L}(\theta, \psi) &= \mathbb{E}\left[\|\hat{Q}_\theta(h, f_\psi(i), a) - (r + \hat{Q}_{\theta'}(h', f_\psi(i'), a'))\|_2^2\right], \\
&\quad \text{where } a' = \arg\max_a \hat{Q}_\theta(h', f_\psi(i'), a) \\
\mathcal{L}(\theta, \omega) &= \mathbb{E}\left[\|\hat{Q}_\theta(h, g_\omega(h), a) - (r + \hat{Q}_{\theta'}(h', g_\omega(h'), a'))\|_2^2\right] \\
&\quad \text{where } a' = \arg\max_a \hat{Q}_\theta(h', g_\omega(h'), a).
\end{aligned}$$

Unlike PROBE, there is no exploration bonus. We use the same embedding scheme as PROBE.

A.4. ELF

We use the ELF implementation from Walsman et al. available at https://github.com/aaronwalsman/impossibly-good. ELF requires the use of action labels from an omniscient expert π^* (with access to the underlying state), which we design by hand in our experiments. It trains a follower policy π_ψ^F that imitates these actions via the cross entropy loss:

$$\mathcal{L}(\psi) = -\mathbb{E}_{a^* \sim \pi^*}\left[\log \pi_\psi^F(a^* \mid o)\right].$$

It also trains a value function for the follower V_ϕ^F via a squared error:

$$\mathcal{L}(\phi) = \mathbb{E}\left[(V_\phi^F(h) - V(h))^2\right].$$

Finally, the explorer policy π_θ^E is trained via PPO (Schulman et al., 2017) to maximize the rewards $\hat{r} = r + V_\phi^F(h_{t+1}) - V_\phi^F(h_t)$. We use the same embedding scheme as PROBE.

A.5. DREAM

DREAM learns an exploration policy π_ϕ and a separate exploration policy π_θ. Both are parameterized as DDQNs. Like PROBE, it also trains an encoder F_ψ and decoder q_ω. The losses are:

$$
\begin{aligned}
\mathcal{L}(\psi) &= \mathbb{E}_i \left[\min(\|f_\psi(i)\|_2^2, K)\right] \\
\mathcal{L}(\omega) &= \mathbb{E}_{h_t} \left[\sum_t \|f_\psi(i_t) - g_\omega(h_t)\|_2^2\right] \\
\mathcal{L}(\phi) &= \mathbb{E}\left[\|\hat{Q}_\phi^{\text{exp}}(h, a) - (r^{\text{exp}} + \gamma \hat{Q}_{\phi'}^{\text{exp}}(h', a'))\|_2^2\right] \\
&\quad \text{where } a' = \arg\max_a \hat{Q}_\phi(h', a) \\
\mathcal{L}(\theta, \psi) &= \mathbb{E}\left[\|\hat{Q}_\theta(h, f_\psi(i), a) - (r + \hat{Q}_{\theta'}(h', f_\psi(i'), a'))\|_2^2\right], \\
&\quad \text{where } a' = \arg\max_a \hat{Q}_\theta(h', f_\psi(i'), a) \\
\mathcal{L}(\theta, \omega) &= \mathbb{E}\left[\|\hat{Q}_\theta(h, g_\omega(h), a) - (r + \hat{Q}_{\theta'}(h', g_\omega(h'), a'))\|_2^2\right] \\
&\quad \text{where } a' = \arg\max_a \hat{Q}_\theta(h', g_\omega(h'), a)
\end{aligned}
$$

and $r^{\text{exp}} = \|f_\psi(i') - g_\omega(h_t)\|_2^2 - \|f_\psi(i') - g_\omega(h_{t+1})\|_2^2 - c$ and c is a hyperparameter. For all of our experiments, we choose $c = 0.1$ following the original work. The losses are similar to PROBE except (1) there is an additional loss term for the explicit exploration policy π_ϕ and (2) the exploitation Q-network $\hat{Q}_\theta$ only trains on the task rewards.

B. Environment Details

B.1. Tiger Door

This problem has 2 possible configurations: either the goal state is at the end of the top hallway or at the end of the bottom hallway. The agent observes its own (x, y) position and a default value of 0. If the agent is on the map cell, the last component of the observation will reveal either 1 or 2, corresponding to the goal at the top and bottom, respectively. The agent can move up, down, left, and right. The agent's reward is -0.1 at each timestep and $+1$ if at the goal. Reaching either the goal or trap cell will terminate the episode. The maximum number of steps is 20.

B.2. Light Dark

In this problem, the agent is randomly initialized in the "dark" part of the room and needs to reach the goal which is at the middle of the dark room. There are 19 possible initial positions (the goal cell is not included). In the dark, the agent observes $(0, 0)$ and in the light, which is on the far right side of the room, the agent observes its true (x, y) position. The agent can move up, down, left, and right. The agent's reward is -0.1 at each timestep and $+1$ if at the goal. Reaching the goal will terminate the episode. The maximum number of steps is 20.

B.3. Map

There are four buses around the middle of the grid, and each teleports the agent to one of the corners. Different configurations have different destinations for the buses, but the destinations are fixed within each episode. Therefore, there are $4!$ possible configurations. The observation includes the agent's current (x, y) position, and an extra component that is by default 0. However, if the agent is at the map state, the observation will reveal an integer corresponding to the index of the current configuration. The agent can move up, down, left, and right, and ride the bus if it is at a stop. The reward is -0.1 at each time-step, and $+1$ if the agent is at the goal. Reaching the goal will terminate the episode. The maximum number of steps is 20.

B.4. Non-stationary Map

This problem builds on Map. Instead of a fixed configuration, the destinations will change after 2 time-steps with probability 0.25 and after 3 time-steps with probability 0.75. Because it takes the agent 2 timesteps to take any bus from the map state, there is a possibility that the destinations change while the agent is taking the bus. In this case, the ideal behavior is to ride the bus back to the middle and visit the map state again. The agent can move up, down, left, and right, and ride the bus if it is at a stop. The reward is -0.05 at each time-step, -0.1 for taking the bus, and $+1$ if the agent is at the goal. Reaching the goal will terminate the episode. The maximum number of steps is 20.

B.5. Overcooked

This problem builds on the Overcooked AI environment. At each timestep, the agent needs to either cook onions or cook tomatoes. The specific dish is determined by the other agent but is unknown to the ego agent. After the first dish is completed, the other agent may potentially choose a new dish to cook. Since there are two possibilities (onion or tomato) for the first and second dish, there are 4 total configurations. The agent's observation includes the current (x, y) position, current direction (i.e., north, east, south, west), type of object being held (can potentially not be holding anything, which then defaults to 0), and an extra component that is by default 0. If the agent visits the top right corner to read the order, the last component of the observation will reveal either 1 or 2, corresponding to the current dish being prepared (onion or tomato). The agent can move north, east, south, and west, and interact with parts of the kitchen (ingredients, pot, dishes, serving station) that it is directly facing. The reward is -0.01 at each timestep, 0.2 for cooking (this bonus is given even if the wrong dish is cooked), and 1 whenever a dish is served. The ideal

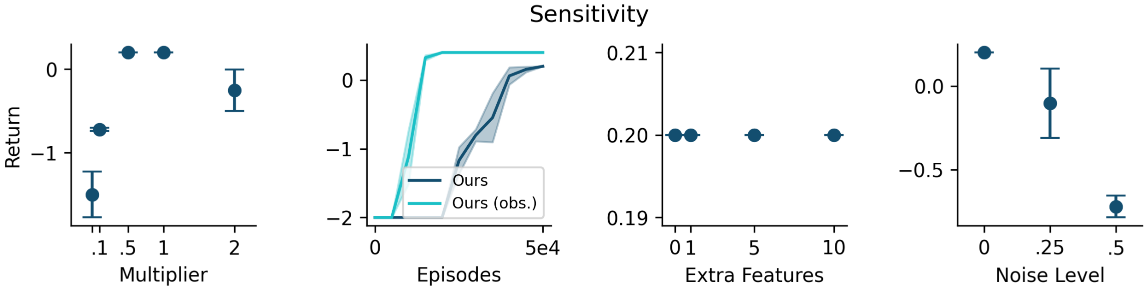

Figure 5. Sensitivity analysis of PROBE. First: PROBE with different weights on the exploration bonus. Second: PROBE defaults to optimal because when the MDP is fully observed. Response of PROBE to extra features (third plot) and to noise (fourth plot) introduced to the PI.

behavior is to first read the order before preparing each of the two dishes. Serving the second dish will terminate the episode. The maximum number of steps is 50.

B.6. Construction

This problem is a 3D visual navigation task. There are two routes, and the agent is randomly initialized at the start of one of these two routes. Each route can either be open or closed for construction. If a route is closed, it will remain closed for the rest of the episode. If a route is open, after every 5 timesteps, there is a probability of 0.7 that the route will close. There is at least one route that will be open for the entire episode, which is unknown to the agent. The agent's observation is a 80×60 RGB image of its egocentric view. Reading the signs at the beginning of each route will indicate whether the route is currently open or closed. If it is closed and the agent walks down the route, a row of construction cones will appear. Either reaching the end of the correct route or walking into the cones will terminate the episode. The agent can turn left by 90 degrees, turn right by 90 degrees, and move forward. The reward is -0.05 at each timestep, -1 for walking into the cones, and 1 for reaching the end of the correct route. The maximum number of steps is 30.

C. Experimental Results

C.1. Sensitivity Analysis

Full information. We next verify that PROBE defaults to a good RL algorithm in the fully-observable setting. We modify the observations of the Tiger Door domain so that the task is fully known even without visiting the map state. Here, we find that the PROBE agent learns the optimal policy more quickly as information gathering is unnecessary (see Fig. 5 (second plot)). Further, it does not need to visit the map state to learn where the goal is, so it solves the task in fewer steps ending with a higher reward.

Excess and noisy privileged information. In most of the earlier experiments, the privileged information precisely described the hidden state. Here, we add random extraneous features to the information vector. Because PROBE applies an information bottleneck on the representation of the privileged information, we find that PROBE is robust to varying amounts of extra information added (see Fig. 5 (third plot)). We also introduce noise to the privileged information and evaluate how PROBE responds to different noise levels. Overall, we find that it is perfectly robust to extra features added, but not as robust to noise, especially when compared to PROBE (FUTURE). Hence, we recommend using the privileged information-free variant of our algorithm when the provided information is suspected to be noisy.

C.2. Additional Results

Information-gathering strategy is expensive to follow. In this experiment, we move the signs farther away from the initial position of the agent, so that reading the signs first is not advantageous to simply walking down the road to check if the road is blocked. Here, the PROBE agent learns to walk down the road to check if it is blocked, rather than reading the signs (see video on the supplementary website).

D. Proof of Theorem 5.2

Our analysis relies on the notion of *information ratio* introduced in Russo & Roy (2014). We adapt it to our setting by defining the information ratio for policy π at episode ℓ as:

$$\Gamma_\ell(\pi) := \frac{\left(\mathbb{E}_\ell\left[\mathcal{J}(\pi^*, \mathcal{E}) - \mathcal{J}(\pi, \mathcal{E})\right]\right)^2}{I_\ell(\widetilde{h}_{T,\ell}; \mathcal{E})},$$

where we use $\mathbb{E}_\ell = \mathbb{E}[\cdot|\widetilde{\mathcal{D}}_\ell]$ and $I_\ell = I(\cdot|\widetilde{\mathcal{D}}_\ell)$ as shorthand.

We utilize the following lemma originally proposed by Russo & Roy (2014):

Lemma D.1. *Let algorithm* $\Pi = \{\pi^\ell\}_{\ell=1}^L$ *deploy policy* π^ℓ *for episode* $\ell \in [L]$ *satisfying*

$$\pi^\ell = \arg\min_\pi \Gamma_\ell(\pi).$$

Then the Bayesian regret bound for the algorithm is

$$\mathcal{BR}(L, \Pi) \leq \sqrt{\mathbb{E}[\Gamma^*] I(\widetilde{\mathcal{D}}_{L+1}; \mathcal{E}) L},$$

where Γ^* is the worst-case information ratio such that $\Gamma_\ell(\pi^\ell) \le \Gamma^*$ for any episode $\ell \in [L]$.

Proof. The proof follows from Theorem 3.1 in Hao & Lattimore (2022). □

For our setting of tabular POMDP with privileged information, we can bound $\mathbb{E}[\Gamma^*]$ and $I(\widetilde{\mathcal{D}}_{L+1};\mathcal{E})$ separately.

Lemma D.2. *Under Assumption 5.1, in a tabular POMDP with privileged information, the worst-case information ratio is bounded as*

$$\mathbb{E}[\Gamma^*] \le 2T^3|\mathcal{S}||\mathcal{A}|\,.$$

Proof. The proof mostly follows from Lemma 3.2 in Hao & Lattimore (2022). However, since we do not have access to the hidden state s_t^ℓ at time t and episode ℓ, we instead replace the POMDP transition dynamics $P^{\mathcal{E}}(\cdot \mid s_h^\ell, a_h^\ell)$ under environment $\mathcal{E}$ with $P^{\mathcal{E}}(\cdot \mid i_h^\ell, a_h^\ell)$ using the privileged information. Since under Assumption 5.1, the two are equivalent, the rest of the proof is unchanged. □

Next, we bound the mutual information:

Lemma D.3. *Under Assumption 5.1, in a tabular POMDP with privileged information, the mutual information can be bounded as*

$$I(\widetilde{\mathcal{D}}_{L+1};\mathcal{E}) \le 2T|\mathcal{S}||\mathcal{I}||\mathcal{A}|\log(LT|\mathcal{S}|)$$

Proof. Again, the proof follows from Lemma 3.3 in Hao & Lattimore (2022). The primary difference is that when constructing a partition such that

$$D_{\mathsf{KL}}\left(P^{\mathcal{E}}(\cdot \mid i,a)||P^{\mathcal{E}'}(\cdot \mid i,a)\right) \le \varepsilon\,,$$

for any environments $\mathcal{E},\mathcal{E}'$, and any i,a,t, we construct a covering over i,a,t instead of s,a,t in the original proof. □

Combining the above lemmas shows that the algorithm that minimizes the information ratio achieves the desired Bayes regret bound. However, the version PROBE that we analyze differs in two key aspects: (1) we consider mutual information between history and the privileged information rather than environment, and (2) we optimize a reward bonus rather than the actual information ratio. We will show that both those changes do not affect the final regret bound.

To handle (1), we will show that the following are equivalent for any policy π:

$$\mathbb{E}_\ell\left[\mathcal{J}(\pi,\mathcal{E})\right] + \alpha I_\ell(\widetilde{h}_T;\mathcal{E}) = \mathbb{E}_\ell\mathbb{E}_\pi\left[\sum_{t=1}^T r(s_t,a_t) + \alpha I(\widetilde{h}_{t-1};i_t)\right]$$

This is because using the chain rule of mutual information, we can show

$$\begin{aligned} I_\ell(\widetilde{h}_T;\mathcal{E}) &= \sum_{t=1}^T \mathbb{E}_\ell\left[I_\ell((i_t,a_t,r_t);\mathcal{E}\mid h_{t-1})\right] \\ &= \sum_{t=1}^T \mathbb{E}_\ell\left[I_\ell(i_t;\mathcal{E}\mid h_{t-1})\right] \\ &\quad + \mathbb{E}_\ell\left[I_\ell(a_t;\mathcal{E}\mid i_t,h_{t-1})\right] \\ &\quad + \mathbb{E}_\ell\left[I_\ell(r_t;\mathcal{E}\mid i_t,a_t,h_{t-1})\right]\,. \end{aligned}$$

As additional notation, let $\bar{\mathcal{E}}_\ell$ denote the mean environment over the posterior over environments $\mathbb{P}(\mathcal{E}=\cdot \mid \widetilde{\mathcal{D}}_\ell)$, such that $\mathbb{E}_\ell\left[P^{\mathcal{E}}(\cdot \mid i,a)\right] = P^{\bar{\mathcal{E}}_\ell}(\cdot \mid i,a)$. Then, we can rewrite the first term as

$$\begin{aligned} &\mathbb{E}_\ell\left[I_\ell(i_t;\mathcal{E}\mid h_{t-1})\right] = \\ &\sum_{(i,a)} \mathbb{P}_{\ell,\pi}^{\bar{\mathcal{E}}_\ell}(i_{t-1}=i,a_{t-1}=a) \\ &\int D_{\mathsf{KL}}\left(P^{\mathcal{E}}(\cdot \mid i,a)||P^{\bar{\mathcal{E}}_\ell}(\cdot \mid i,a)\right)d\mathbb{P}_\ell(\mathcal{E}) \\ &= \int \mathbb{E}_\pi\left[D_{\mathsf{KL}}\left(P^{\mathcal{E}}(\cdot \mid i,a)||P^{\bar{\mathcal{E}}_\ell}(\cdot \mid i,a)\right)\right]d\mathbb{P}_\ell(\mathcal{E}) \\ &= \mathbb{E}_\ell\mathbb{E}_\pi\left[D_{\mathsf{KL}}\left(P^{\mathcal{E}}(\cdot \mid i,a)||P^{\bar{\mathcal{E}}_\ell}(\cdot \mid i,a)\right)\right] \\ &= \mathbb{E}_\ell\mathbb{E}_\pi\left[I(\widetilde{h}_{t-1};i_t)\right] \end{aligned}$$

In addition, we see that the second and third terms are 0 because the action taken is purely a function of π and not the underlying environment, and reward is deterministic. This means that the desired equivalence holds.

Finally, what remains is tackling (2). Using the AM-GM inequality for any policy π, we have

$$\begin{aligned} &\frac{\mathbb{E}_\ell\left[\mathcal{J}(\pi^*,\mathcal{E}) - \mathcal{J}(\pi,\mathcal{E})\right]\sqrt{\alpha I_\ell(\widetilde{h}_{T,\ell};\mathcal{E})}}{\sqrt{\alpha I_\ell(\widetilde{h}_{T,\ell};\mathcal{E})}} \le \\ &\frac{\left(\mathbb{E}_\ell\left[\mathcal{J}(\pi^*,\mathcal{E}) - \mathcal{J}(\pi,\mathcal{E})\right]\right)^2}{2\alpha I_\ell(\widetilde{h}_{T,\ell};\mathcal{E})} + \frac{\alpha}{2}I_\ell(\widetilde{h}_{T,\ell};\mathcal{E})\,. \end{aligned}$$

This means that

$$\mathbb{E}_\ell\left[\mathcal{J}(\pi^*,\mathcal{E}) - \mathcal{J}(\pi,\mathcal{E})\right] - \frac{\alpha}{2}I_\ell(\widetilde{h}_{T,\ell};\mathcal{E}) \le \frac{\Gamma_\ell(\pi)}{2\alpha}$$

Rearranging and choosing $\alpha = \sqrt{6L\mathbb{E}[\Gamma^*]/I(\widetilde{\mathcal{D}}_{L+1};\mathcal{E})}$ gives us the original regret bound, as desired.