ContraBAR: Contrastive Bayes-Adaptive Deep RL

Era Choshen [1] **Aviv Tamar** [1]

Abstract

In meta reinforcement learning (meta RL), an agent seeks a Bayes-optimal policy – the optimal policy when facing an unknown task that is sampled from some known task distribution. Previous approaches tackled this problem by inferring a *belief* over task parameters, using variational inference methods. Motivated by recent successes of contrastive learning approaches in RL, such as contrastive predictive coding (CPC), we investigate whether contrastive methods can be used for learning Bayes-optimal behavior. We begin by proving that representations learned by CPC are indeed sufficient for Bayes optimality. Based on this observation, we propose a simple meta RL algorithm that uses CPC in lieu of variational belief inference. Our method, *ContraBAR*, achieves comparable performance to state-of-the-art in domains with state-based observation and circumvents the computational toll of future observation reconstruction, enabling learning in domains with image-based observations. It can also be combined with image augmentations for domain randomization and used seamlessly in both online and offline meta RL settings.

1. Introduction

In meta reinforcement learning (meta RL), an agent learns from a set of training tasks how to quickly solve a new task, sampled from a similar distribution as the training set (Finn et al., 2017; Duan et al., 2016). A formal setting for meta RL is based on the Bayesian RL formulation, where a task corresponds to a particular Markov decision process (MDP), and there exists some *prior* distribution over MDPs (Humplik et al., 2019; Zintgraf et al., 2020; Rakelly et al., 2019). Under this setting, the *optimal* meta RL policy is well defined, and is often referred to as a *Bayes-optimal* policy (Ghavamzadeh et al., 2016).

In contrast to the single MDP setting, where an optimal policy can be Markovian – taking as input the current state and outputting the next action, the Bayes-optimal policy must take as input *the whole history* of past states, actions, and rewards, or some sufficient statistic of it (Bertsekas, 1995). A popular sufficient statistic is the *belief* – the posterior probability of the MDP parameters given the observed history. For small MDPs, the belief may be inferred by directly applying Bayes rule, and approximate dynamic programming can be used to calculate an approximately Bayes-optimal policy (Ghavamzadeh et al., 2016). However, this approach quickly becomes intractable for large or continuous MDPs. Recently, several studies proposed to scale up belief inference using deep learning, where the key idea is to leverage a variational autoencoder (VAE, Kingma & Welling 2013) formulation of the problem, in which the posterior is approximated using a recurrent neural network (Zintgraf et al., 2020; Humplik et al., 2019). While this approach has demonstrated impressive results on continuous control benchmarks (Zintgraf et al., 2020; Humplik et al., 2019; Dorfman et al., 2021), it also has some limitations. Training a VAE is based on a *reconstruction* loss, in this case, predicting the future observations given the current history, which can be difficult to optimize for visually rich observations such as images. Furthermore, variational algorithms such as VariBAD (Zintgraf et al., 2020) reconstruct entire trajectories, restricting application to image-based domains due to memory limitations.

As an alternative to VAEs, contrastive learning has shown remarkable success in learning representations for various domains, including image recognition and speech processing (Chen et al., 2020; Fu et al., 2021b; Han et al., 2021). Rather than using a reconstruction loss, these approaches learn features that discriminate between similar observations and dissimilar ones, using a contrastive loss such as the InfoNCE in contrastive predictive coding (CPC, Oord et al. 2018). Indeed, several recent studies showed that contrastive learning can learn useful representations for image based RL (Laskin et al., 2020; 2022), outperforming representations learned using VAEs. Furthermore, Guo et al. (2018) showed empirically that in partially observed MDPs, representations learned using CPC (Oord et al., 2018) are correlated with the belief. In this work, we further investigate contrastive learning for meta RL, henceforth termed

[1] Technion, Haifa, Israel. Correspondence to: Era Choshen <erachoshen@campus.technion.ac.il>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

CL meta RL, and aim to establish it as a principled and advantageous alternative to the variational approach.

Our first contribution is a proof that, given certain assumptions on data collection and the optimization process of CPC, representations learned using a variant of CPC are indeed a sufficient statistic for control, and therefore suffice as input for a Bayes-optimal policy. Our second contribution is a bound on the suboptimality of a policy that uses an *approximate* sufficient statistic, learned by CPC, in an iterative policy improvement scheme where policies between iterations are constrained to be similar. This result relaxes the assumptions on the optimization and data collection in the first proof. Building on this result, we propose a simple meta RL algorithm that uses a CPC based representation to learn a sufficient statistic. Our third contribution is an empirical evaluation of our method that exposes several advantages of the contrastive learning approach. In particular, we show that: (1) For state-based observations, CL meta RL is on par with the state-of-the-art VariBAD (Zintgraf et al., 2020) (2) For image-based observations, CL meta RL significantly outperforms the variational approach, and is competitive with RNN based methods (Duan et al., 2016) (3) In contrast to the variational approach, CL Meta RL is compatible with image augmentations and domain randomization. (4) Our method works well in the online and offline meta RL setting. Overall, our results establish CL meta RL as a versatile and competitive approach to meta RL.

2. Background and Problem Formulation

In this section we present our problem formulation and relevant background material.

2.1. Meta RL and POMDPs

We define a Markov Decision Process (MDP) (Bertsekas, 1995) as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is the transition kernel and $\mathcal{R}$ is the reward function. In meta RL, we assume a distribution over tasks, where each task is an MDP $\mathcal{M}_i = (\mathcal{S}, \mathcal{A}, \mathcal{P}_i, \mathcal{R}_i)$, where the state and action spaces are shared across tasks, and $\mathcal{P}_i, \mathcal{R}_i$ are task specific and drawn from a task distribution, which we denote $\mathcal{D}(\mathcal{P}, \mathcal{R})$. At a given time t, we denote by $(s_0, a_0, r_0, s_1, a_1, r_1, \dots, s_t) = h_t \in \Omega_t$ the current history, where Ω_t is the space of all state-action-reward histories until time t. Our aim in meta RL is to find a policy $\pi = \{\pi_0, \pi_1, \dots\}$, where $\pi_t : \Omega_t \to \mathcal{A}$, which maximizes the following objective:

$$\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t \right],$$

where the expectation $\mathbb{E}_\pi$ is taken over the transitions $s_{t+1} \sim \mathcal{P}(\cdot | s_t, a_t)$, the reward $r_t = \mathcal{R}(s_t, a_t)$, the actions $a_t \sim \pi(\cdot | h_t)$ and the uncertainty over the MDP parameters $\mathcal{P}, \mathcal{R} \sim \mathcal{D}(\mathcal{P}, \mathcal{R})$. We assume a bounded reward $r_t \in [-R_{max}, R_{max}], R_{max} > 0$ with probability one.

Meta RL is a special case of the more general Partially Observed Markov-Decision Process (POMDP), which is an extension of MDPs to partially observed states. In the POMDP for meta RL, the unobserved variables are $\mathcal{P}, \mathcal{R}$, and they do not change over time. We define Ω_t for POMDPs as above, except that states are replaced by observations according to the distribution $o_{t+1} \sim U(o_{t+1} | s_{t+1}, a_t)$. As shown in Bertsekas (1995), the optimal policy for a POMDP can be calculated using backwards dynamic programming for every possible $h_t \in \Omega_t$. However, as explained in Bertsekas (1995) this method is computationally intractable in most cases as Ω_t grows exponentially with t.

2.2. Information States and BAMDPs

Instead of the intractable space of histories, *sufficient statistics* can succinctly summarize all the necessary information for optimal control. One popular sufficient statistic is the posterior state distribution or *belief* $P(s_t | h_t)$. Conditions for a function to be a sufficient statistic, also termed *information state*, were presented by Subramanian et al. (2020) and are reiterated here for completeness:

Definition 2.1 (Information State Generator)**.** Let $\{\mathcal{Z}_t\}_{t=1}^T$ be a pre-specified collection of Banach spaces. A collection $\{\sigma_t : \Omega_t \to \mathcal{Z}_t\}_{t=1}^T$ of history compression functions is called an information generator if the process $\{Z_t\}_{t=1}^T$ satisfies the following properties, where $h_t \in \Omega_t$, and $\sigma_t(h_t) = Z_t \in \mathcal{Z}_t$:

P1 For any time t and for any $h_t \in \Omega_t, a_t \in \mathcal{A}$ we have:

$$\mathbb{E}\left[r_t | h_t, a_t\right] = \mathbb{E}\left[r_t | Z_t = \sigma_t(h_t), a_t\right].$$

P2 For any time t, and for any $h_t \in \Omega_t, a_t \in \mathcal{A}$, and any Borel subset B of Z_{t+1} we have:

$$P\left(B \in Z_{t+1} | h_t, a_t\right) = P\left(B \in Z_{t+1} | Z_t = \sigma_t(h_t), a_t\right).$$

Intuitively, information states compress the history without losing predictive power about the next reward, or the next information state.

To solve a POMDP, one can define a Bayes-Adaptive MDP (BAMDP)– an MDP over the augmented state space of $\mathcal{S} \times \mathcal{B}$, where $\mathcal{B} = \{\mathcal{Z}_t\}_{t=1}^T$ is the space of the information state. This idea was introduced by Duff (2002) for the belief. Here, we use the term BAMDP more generally, referring to any information state. The optimal policies for BAMDPs are termed Bayes-optimal and optimally trade-off between exploration and exploitation, which is essential for maximizing online return during learning. Unfortunately, in most cases computing the Bayes-optimal policy is intractable because the augmented space is continuous and high-dimensional. Zintgraf et al. (2020) proposed to approximate the Bayes-optimal policy by using deep neural networks to learn an information state (belief), and conditioning an RL agent on the learned augmented space; here we follow this approach.

3. Related Work

Our focus in this work is learning a Bayes-optimal policy for meta RL. We recapitulate the current approaches to meta RL with a focus on approaches that potentially yield Bayes-optimal policies.

The methods in (Finn et al., 2017; Grant et al., 2018; Nichol et al., 2018; Rothfuss et al., 2018; Clavera et al., 2018) learn neural network policies that can quickly be fine-tuned to new tasks at test time via gradient updates. These methods do not optimize for Bayes-optimal behavior, and typically exhibit significantly suboptimal test-time adaptation.

A different approach is to learn an agent that directly infers the task at test time, and conditions the policy based on the inferred task. Typically, past interactions of the agent with the environment are aggregated to a latent representation of the task. Rakelly et al. (2019); Fakoor et al. (2019) follow a posterior-sampling approach, which is not Bayes-optimal (Zintgraf et al., 2020); in this work we focus on methods that can achieve Bayes-optimality. Duan et al. (2016); Wang et al. (2016) propose memory-based approaches, which Ortega et al. (2019) proves to approximate Bayes-optimal agents. Zintgraf et al. (2020; 2021); Dorfman et al. (2021) also approximate Bayes-optimal agents with a history-based representation, using a variational approach. Humplik et al. (2019) learn an approximately Bayes-optimal agent, where privileged information – a task descriptor – is used to learn a sufficient statistic. We explore an alternative approach that lies at the intersection of meta RL and contrastive learning. Different from memory-based methods such as RL^2 (Duan et al., 2016), and similarly to VariBAD, we learn a history based embedding *separately* from the policy. However, unlike variational methods, we learn the task representation using contrastive learning.

Contrastive learning has been used to learn representations for input to a meta RL policy. FOCAL (Li et al., 2020b) uses distance metric learning to learn a deterministic encoder of transition tuples to perform offline RL. They operate under the relatively restrictive assumption that each transition tuple (s, a, s', r) is uniquely identified by a task. The authors followed up with FOCAL++, in which batches of transition tuples (not necessarily from the same trajectory) are encoded to a representation that is optimized with MoCo (He et al., 2020), a variant of CPC, alongside an intra-task attention mechanism meant to robustify task inference (Li et al., 2021). The MBML method in (Li et al., 2020a) proposes an offline meta RL method that uses the triplet loss to learn embeddings of batches of transition tuples from the same task, with the same probabilistic and permutation-invariant architecture of Rakelly et al. (2019). Wang et al. (2021) propose embedding windows of transition tuples as probabilistic latent variables, where the windows are cropped from different trajectories. The embeddings are learned with MoCO (He et al., 2020) by contrasting them in probabilistic metric space, where positive pairs are transition windows that come from the same batch. The algorithm is presented as a general method to learn representations for context-based meta RL algorithms, but in practice all results are shown with PEARL (Rakelly et al., 2019). In a similar line of work, Fu et al. (2021a) encode batches of transitions as a product of Gaussian factors and contrast the embeddings with MoCO (He et al., 2020), with positive pairs being embedded transition batches from the same *task*, as opposed to the same trajectory as in Wang et al. (2021). As in Wang et al. (2021), results are shown with a posterior sampling meta RL algorithm. While we also investigate contrastive learning for meta RL, we make an important distinction: all of the works above embed *transition tuples and not histories*, and therefore cannot represent information states, and cannot obtain Bayes-optimal behavior. In contrast, in our work, we draw inspiration from Guo et al. (2018), who used a glass-box approach to empirically show that contrastive learning *can* be used to learn the belief in a POMDP. We cast this idea in the Bayesian-RL formalism, and show both theoretically and empirically, that contrastive learning can be used to learn Bayes-optimal meta RL policies.

4. Method

In this section we show how to use contrastive learning to learn an information state representation of the history, and use it as input to an RL agent. We give a brief description of CPC (Oord et al., 2018) followed by our meta RL algorithm. We then prove that our method does indeed learn an information state.

4.1. Contrastive Predictive Coding

CPC (Oord et al., 2018) is a contrastive learning method that uses noise contrastive estimation (Gutmann & Hyvärinen, 2010) to discriminate between positive future observations o^+_{t+k}, where t is the current time step, and negative observations o^-_{t+k}. First, an encoder g generates an embedding for each observation in a sequence of observations from a trajectory τ until time t, $\{z_i = g(o_i)\}_{i=1}^t$. Second, an autoregressive model g_{AR} summarizes $z_{\leq t}$, the past t observations in latent space, and outputs a latent c_t. The model is trained to discriminate between future observations o^+_{t+k} and K negative observations $\left\{o^{-,i}_{t+k}\right\}_{i=1}^K$ given c_t. Given

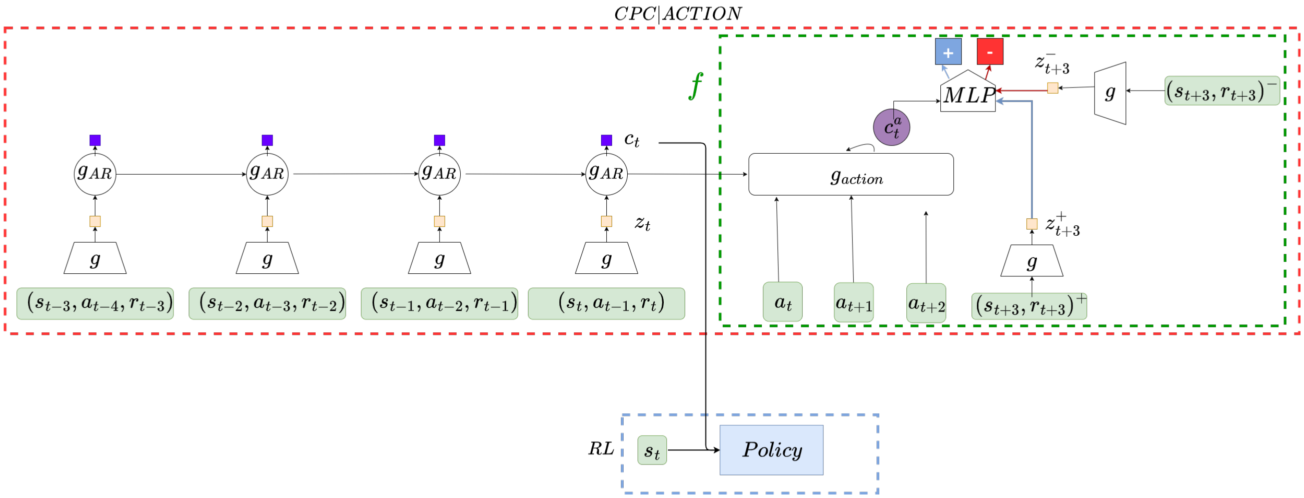

Figure 1: ContraBAR architecture. Red box: similarly to CPC, g_{AR} is autoregressively applied to past observation embeddings $(z_t, z_{t-1}, \dots)$ to yield the current representation c_t. Green box: unlike CPC, the MLP f receives as input, in addition to a positive/negative future embedding, the output of an action GRU that process c_t and the future action sequence. Blue box: the representation c_t is the information state, which is concatenated with the state s_t and input to the RL policy.

a set $X = \{o^+_{t+k}, o^{-,1}_{t+k}, \dots, o^{-,K}_{t+k}\}$ containing one positive future observation sampled according to $P(o^+_{t+k}|c_t)$ and K negative observations sampled from a proposal distribution $P(o^-_{t+k})$, the InfoNCE loss is:

$$\mathcal{L}_{InfoNCE} =$$

$$-\mathbb{E}_X\left[\log \frac{\exp\left(f\left(c_t, o^+_{t+k}\right)\right)}{\exp\left(f\left(c_t, o^+_{t+k}\right)\right) + \sum_{i=1}^K \exp\left(f\left(c_t, o^{-,i}_{t+k}\right)\right)}\right],$$

where f is a learnable function that outputs a similarity score. The model components f, g, g_{AR} are learned by optimizing the loss $\mathcal{L}_{InfoNCE}$.

4.2. ContraBAR Algorithm

We will now introduce our CPC based meta RL algorithm, which is depicted in Figure 1, and explain how CPC is used to learn a latent representation of the history.

We begin by noting that we use the term observation throughout the text, in line with Oord et al. (2018), however in our case the meaning is state, reward and action $o_t = \{s_t, r_{t-1}, a_{t-1}\}$ when talking about "observation history", and state and reward $o_{t+k} = \{s_{t+k}, r_{t+k-1}\}$ when talking about "future observations". We would like to learn an embedding of the observation history, c_t, that will contain relevant information for decision making. The CPC formulation seems like a natural algorithm to do this – for a given trajectory τ of length T collected from some unknown MDP $\mathcal{M}$, we use an embedding of its observation history until time $t < T$, c_t, to learn to discriminate between future observations from the trajectory τ and random observations from other trajectories $\tau_j \neq \tau$. This means that c_t encodes relevant information for predicting the future system states, and consequently information regarding the MDP $\mathcal{M}$ from which τ was collected.

The CPC formulation described above is based on predicting future states in an uncontrolled system without rewards. We now modify it to learn a sufficient statistic for meta RL. We assume that data is collected at each training iteration m by some data collection policy π_m and added to a replay buffer $\mathcal{D} = \{\tau_i\}_{i=1}^N$ containing trajectories from previous data collection policies $\{\pi_1, \dots, \pi_{m-1}\}$; we note that length of the trajectories may vary. At each learning iteration, a batch of M trajectories is sampled, and for each trajectory and time t the negative observations are sampled from the remaining $M-1$ trajectories in the batch. As in CPC, we define c_t to be a function of the observation history until time t, but we add to f as input the future $k-1$ actions, as in a controlled system the future observation $o_{t+k} = (s_{t+k}, r_{t+k-1})$ depends on the controls. Our f can therefore now be written as $f(c_t, o_{t+k}, a_{t:t+k-1})$. We implement this modification as in Guo et al. (2018) by means of an additional autoregressive component, a GRU g_{action} that receives actions as input and takes c_t as its initial hidden state.

Given the adjustments described above, each batch B used as input to our algorithm contains the following: (1) The observation history until time t in some trajectory τ (2) Future observations from time $t+k$ (3) Observations o^-_{t+k} from the remaining $M-1$ trajectories sampled from $\mathcal{D}$.

We rewrite the InfoNCE loss for the meta RL setting explicitly; For ease of notation we mark $f(c_t, o^{+}_{t+k}, a_{t:t+k-1})$ as f^{+} and $f(c_t, o^{-,i}_{t+k}, a_{t:t+k-1})$ as $f^{-,i}$:

$$\mathcal{L}_M = \mathbb{E}_B \left[\log \frac{\exp(f^{+})}{\exp(f^{+}) + \sum_{i=1}^{K} \exp\left(f^{-,i}\right)} \right], \quad (1)$$

where the expectation is over the batches of positive and negative observations sampled from $\mathcal{D}$, as described above.

4.3. Learning Information States with CPC

We now show that integrating contrastive learning with meta RL is a fundamentally sound idea. We shall prove that our algorithm presented in Section 4.2 learns a representation of the history that is an information state, by showing that the latent encoding satisfies the properties of an information state **P1, P2** as defined by Subramanian et al. (2020) and reiterated above in Section 2.

We first define the notion of a "possible" history, which we use in Assumption 4.2.

Definition 4.1 (Possible history)**.** Let P_M denote a probability distribution over MDPs and let $P_{m,\pi}(h_t)$ be the probability of observing history h_t under policy π in MDP m. We say that h_t is a possible history if there exists a policy π and an MDP m such that $P_M(m) > 0$ and $P_{m,\pi}(h_t) > 0$.

Next, we make the following assumption, which states that the policy collecting the data covers the state, reward and action space.

Assumption 4.2. Let the length of the longest possible history be T. Let h_t, where $t \leq T$, be a history and let $P_{\mathcal{D}}(h_t)$ denote the probability of observing a history in the data $\mathcal{D}$. If h_t is a possible history, then $P_{\mathcal{D}}(h_t) > 0$.

Assumption 4.2 is necessary to claim that the learned CPC representation is a sufficient statistic *for every* possible history. In Section 4.4 we discuss a relaxation of this assumption, using *approximate* information states.

Theorem 4.3. *Let Assumption 4.2 hold. Let g^*, g^*_{AR}, f^* jointly minimize $\mathcal{L}_M(g, g_{AR}, f)$. Then the context latent representation $c_t = g^*_{AR}(z_{\leq t})$ satisfies conditions **P1, P2** and is therefore an information state.*

The full proof is provided in Appendix A; we next provide a sketch. The main challenge in our proof lies in proving the following equality:

$$P(s_{t+1}, r_t | h_t, a_t) = P(s_{t+1}, r_t | c_t, a_t). \quad (2)$$

Given the equality in Equation (2), proving **P1,P2** is relatively straightforward. We prove Equation (2) by expanding the proof in (Oord et al., 2018), which shows that the InfoNCE loss upper bounds the negative mutual information between o^{+}_{t+k} and c_t (in the CPC setting). In our case, we show that

$$\mathcal{L}_M \geq \log(M-1) - I(s_{t+1}, r_t; c_t | a_t), \quad (3)$$

where $I(\cdot;\cdot)$ denotes mutual information. Thus, by minimizing the loss in Equation (1), we maximize the mutual information $I(s_{t+1}, r_t; c_t|a_t)$. Due to the Markov property of the process, the mutual information in (3) cannot be greater than $I(s_{t+1}, r_t; h_t|a_t)$, which leads to Equation 2.

4.4. Learning Approximate Information States with CPC

We next investigate a more practical setting, where there may be errors in the CPC learning, and the data does not necessarily satisfy Assumption 4.2. We aim to relate the CPC error to a bound on the suboptimality of the resulting policy. In this section, we consider an iterative policy improvement algorithm with a similarity constraint on consecutive policies, similar to the PPO algorithm we use in practice (Schulman et al., 2017). We shall bound the suboptimality of policy improvement, when data for training CPC is collected using the previous policy, denoted π_k.

In light of Eq. 3, we assume the following error due to an imperfect CPC representation:

Assumption 4.4. There exists an ϵ such that for every $t \leq T$, $I(s_{t+1}, r_t; c_t|a_t) \geq I(s_{t+1}, r_t; h_t|a_t) - \epsilon$, where the histories are distributed according to policy π_k.

The next theorem provides our main result.

Theorem 4.5. *Let Assumption 4.4 hold for some representation c_t. Consider the distance function between two distributions $D(P_1(x), P_2(x)) = \max_x |P_1(x)/P_2(x)|$. We let $\hat{r}(c_t, a_t) = \mathbb{E}[r_t|c_t, a_t]$ and $\hat{P}(c'|c_t, a_t) = \mathbb{E}[\mathbf{1}(c_{t+1} = c')|c_t, a_t]$ denote an approximate reward and transition kernel, respectively. Define the value functions*

$$\hat{Q}_t(c_t, a_t) = \hat{r}(c_t, a_t) + \sum_{c_{t+1}} \hat{P}(c_{t+1}|c_t, a_t)\hat{V}_{t+1}(c_{t+1})$$

$$\hat{V}_t(c_t) = \max_{\pi: D(\pi(c_t), \pi_k(c_t)) \leq \beta} \sum_a \pi(a)\hat{Q}_t(c_t, a), \quad (4)$$

for $t \leq T$, and $\hat{V}_T(c_T) = 0$, and the approximate optimal policy

$$\hat{\pi}(c_t) \in \operatorname*{argmax}_{\pi: D(\pi, \pi_k(c_t)) \leq \beta} \sum_a \pi(a)\hat{Q}_t(c_t, a). \quad (5)$$

Let the optimal policy $\pi^*(h_t)$ be defined similarly, but with h_t replacing c_t in (4) *and* (5)*. Then we have that*

$$\mathbb{E}^{\pi^*}\left[\sum_{t=0}^{T-1} r(s_t, a_t)\right] - \mathbb{E}^{\hat{\pi}}\left[\sum_{t=0}^{T-1} r(s_t, a_t)\right] \leq$$
$$\epsilon^{1/3} R_{max} T^2 (\sqrt{2} + 4\beta^T).$$

The dynamic programming recurrence in Equation (4) defines the optimal policy that is conditioned on c_t (and not h_t), and is restricted to be β-similar to the previous policy π_k. The theorem bounds the loss in performance of such a policy compared to a policy that is conditioned on the full history (yet still restricted to be β-similar to π_k). The proof of Theorem 4.5 builds on the idea of an approximate information state (Subramanian et al., 2020) and is detailed in Appendix A.

4.5. ContraBAR Architecture

We now describe several design choices in our ContraBAR implementation.

History Embedding We now describe the specific architecture used to implement our algorithm, also depicted in Figure 1. We use a non-linear encoder to embed a history of actions, rewards and states and run it through a GRU to generate the hidden state for the current time-step c_t. The latent c_t is then used to initialize the action-gru g_{action}, which is fed future actions as input – the resulting hidden state is then concatenated with either a positive observation-reward pair, or a negative one and used as input to a projection head that outputs a score used in Equation (1).

We note that given a random sampling of negative observations, the probability of sampling a positive and negative observation that share the same state is low. Consequently, for environments where s_{t+k} can be estimated via $s_t, a_t, \dots, a_{t+k}$ without h_t, c_t need only encode information regarding s_t to allow the action-gru to learn to distinguish between positive and negative observations. This renders c_t uninformative about the reward and transition functions and thus unhelpful for optimal control. An example of this is a set of deterministic environments that differ only in reward functions. The action-gru can learn to predict s_{t+k} via $s_t, a_t, \dots, a_{t+k}$, only requiring c_t to encode information regarding s_t and *not* the reward function. One way to circumvent this is hard negative mining, i.e using negative samples that are difficult to distinguish from the positive ones. Another solution, relevant for the case of varying reward functions, is to generate a negative observation by taking the state and action from the positive observation and recalculating the reward with a reward function sampled from the prior. In practice, we found that a simple alternative is to omit the action-gru. This prevents the easy estimation of s_{t+k} and requires c_t to encode information regarding the reward and transition function. We found this worked well in practice for the environments we ran experiments on, including those with varying transitions. We expand on these considerations in Appendix D.

RL Policy The history embedding portion of the algorithm described above is learned separately from the policy and can be done online or offline. The policy, which can be trained with an RL algorithm of the user's choice, is now conditioned on the current state s_t as well as c_t – the learned embedding of h_t. We chose to use PPO (Schulman et al., 2017) for the online experiments and SAC (Haarnoja et al., 2018) for the offline experiment – in line with VariBAD and BOReL (Dorfman et al., 2021).

5. Experiments

In our experiments, we shall demonstrate that (1) ContraBAR learns approximately Bayes-optimal policies (2) ContraBAR is on par with SOTA for environments with state inputs (3) ContraBAR scales to image-based environments (4) Augmentations can be naturally incorporated into ContraBAR and (5) ContraBAR can work in the offline setting.

We compare ContraBAR to state-of-the-art approximately Bayes-optimal meta RL methods. In the online setting, we compare against VariBAD (Zintgraf et al., 2020), RL^2 (Duan et al., 2016), and the recent modification of RL^2 by Ni et al. (2022) which we refer to as RMF (recurrent model-free). In the offline setting, we compare with BOReL (Dorfman et al., 2021). Zintgraf et al. (2020) and Dorfman et al. (2021) already outperform posterior sampling based methods such as PEARL (Rakelly et al., 2019), therefore we do not include such methods in our comparison. Finally, we note that using VariBAD (Zintgraf et al., 2020) with image-based inputs is currently computationally infeasible due to memory constraints, and as such we did not use it as a baseline – we explain this issue further in Appendix E. Other variational approaches, which require a reconstruction of the future observations, are subject to similar memory constraints. Instead, we compared our algorithm against RL^2 (Duan et al., 2016), which works with images. We evaluate performance similarly to Zintgraf et al. (2020), by evaluating per episode return for 5 consecutive episodes with the exception of the offline setting where we adapted our evaluation to that of BOReL.

5.1. Qualitative Near Bayes-Optimal Behavior

We begin with a qualitative demonstration that ContraBAR can learn near Bayes-optimal policies. As calculating the exact Bayes-optimal policy is mostly intractable, we adopt the approach of Dorfman et al. (2021): for deterministic domains with a single sparse reward, the Bayes-optimal solution is essentially to search all possible reward locations so as to maximally reduce uncertainty, and then go directly to the goal in subsequent episodes. Thus, we can identify whether a policy is approximately Bayes-optimal by inspecting its trajectory. Figure 2 displays rollouts from a trained policy in the Gridworld and Semi-Circle domains, demonstrating near Bayes-optimal behavior similar to VariBAD (Zintgraf et al., 2020).

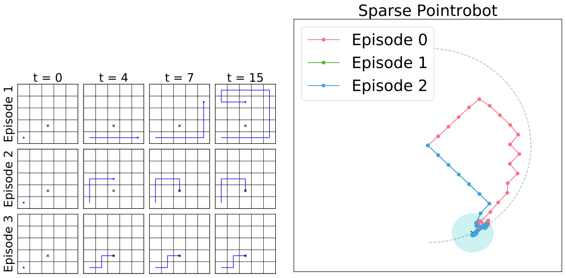

Figure 2: **Left:** Qualitatively Bayes-Optimal behavior on gridworld. The agent methodically searches the grid, reducing uncertainty about unexplored cells in the second episode and updating to a shorter path in the subsequent episode. **Right:** Qualitatively Bayes-Optimal behaviour on Sparse Semi-Circle. The agent scans the semi-circle until the goal has been found and all uncertainty reduced. In subsequent episodes the agent goes straight to the goal.

5.2. Results for Problems with State Observations

We compare ContraBAR with VariBAD and RMF, the current state-of-the-art on MuJoCo locomotion tasks (Todorov et al., 2012), commonly used in meta RL literature. We use the environments considered in Zintgraf et al. (2020), namely the Ant-Dir, AntGoal, HalfCheetahDir, HalfCheetahVel, Humanoid and Walker environments. Figure 3 shows competitive performance with the current SOTA on all domains. Note that rewards in these environments are dense, so in principle, the agent only needs a few exploratory actions to infer the task by observing the rewards it receives. Indeed, we see that ContraBAR is able to quickly adapt within the first episode, with similar performance in subsequent episodes.

5.3. Scaling Belief to Image-Based Inputs

We show that ContraBAR can scale to image domains, which are computationally expensive, by running our algorithm on three image-based domains with varying levels of difficulty and sources of uncertainty: (1) Reacher-Image – a two-link robot reaching an unseen target located somewhere on the diagonal of a rectangle, with sparse rewards (2) Panda Reacher – a Franka Panda robot tasked with placing the end effector at a goal on a 2d semi-circle, where the vertical position of the goal (z coordinate) is fixed; adapted from the Reacher task in Panda Gym (Gallouédec et al., 2021) (3) Panda Wind – The same environment as Panda Reacher, except that the transitions are perturbed with Gaussian noise sampled separately for each task. For a more detailed description of each environment see Appendix B.

Image-Based Reward: For our image-based experiments, we found that learning in image-based domains with sparse reward was difficult when the reward was embedded separately (as in the state observation domains), and concatenated with the image embedding. We hypothesized that this might be an issue of differing scales between the scalar rewards and image inputs, but we observed that standard normalization techniques such as layer norm (Ba et al., 2016) did not help. Instead, we opted for a different approach that embeds the reward as an explicit part of the image. To implement this idea, we exploited the fact that in all our domains, the reward is sparse and binary, and we add a colored strip to a fixed place in the image when non-zero reward is received. Extending this idea to non-binary reward is possible, for example, by controlling the color of the strip.

Our results are displayed in Figure 4. For the Reacher environment, ContraBAR is slightly outperformed by RL^2, whereas in Panda Reacher and Panda Reacher Wind ContraBAR outperform RL^2 by a large margin. Notice that in contrast to the dense reward domains of Section 5.2, in these sparse reward tasks the agent gains by exploring for the goal in the first iteration. Evidently, the plots show significantly higher reward in the second episode onward.

Glass-box Approach To further validate that our algorithm learns a sound belief representation, we follow a glass-box approach similar to that of Guo et al. (2018). First, we used ContraBAR to learn an information state for the Panda Reacher environment. Second, we use the trained agent to create a dataset of trajectories, including the agent's belief at each time step of every trajectory. We then trained an MLP-based binary classifier, which takes (x, y) and the information state c_t as input and predicts whether the goal in the trajectory is indeed (x, y). In Figure 5 we see the visualization of the classifier's prediction at different points along the trajectory; We see that the predictions coincide with the belief we expect the agent to hold at each step, thus validating the soundness of our belief representation.

5.4. ContraBAR with Domain Randomization

Despite the high fidelity of modern simulators, when deployed in the real-world, image-based algorithms learned in simulation can only be accurate up to the differences between simulation and reality – the sim-to-real gap. This motivates us to learn a belief representation that is robust to such differences, and in the following we will show that our algorithm can indeed learn such an information state. Robustification to irrelevant visual properties via random modifications is termed *domain randomization* (Tobin et al., 2017). We employ domain randomization in a similar fashion to Rabinovitz et al. (2021) wherein we modify the past and future observations (without the rewards) in the trajectories with a mapping $\mathcal{T} : \mathcal{S} \to \mathcal{S}$ that randomly shifts the RGB channels of the images. These modified trajectories are used to learn the history embedding c_t, with the hope that it will be invariant to different color schemes in the environment. We show the strength of such modifications by

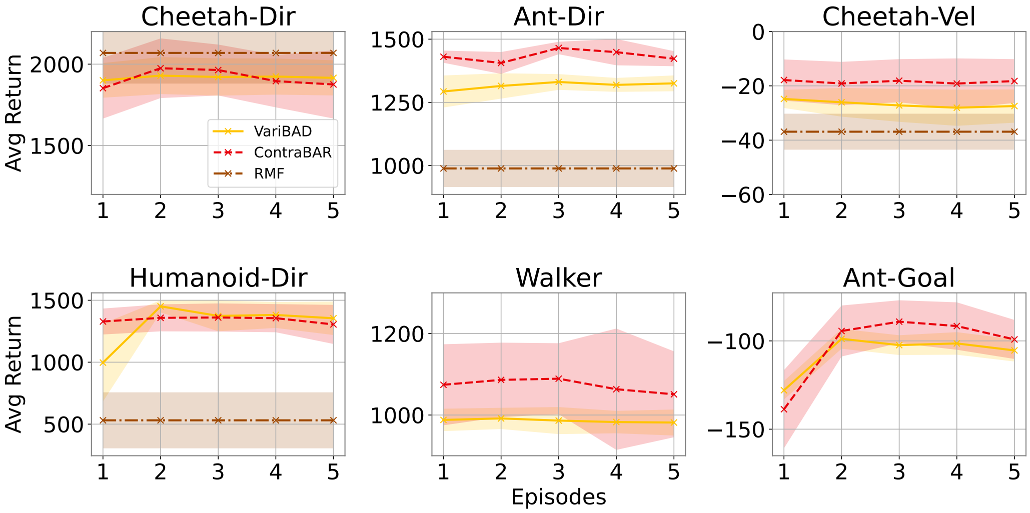

Figure 3: Average test performance on six different MuJoCo environments, trained separately with 10 seeds per MuJoCo environment per method - as in (Zintgraf et al., 2020). The meta-trained policies are rolled out for 5 episodes to show how they adapt to the task. Values shown are averages across tasks (95% confidence intervals shaded). We show competitive performance, surpassing VariBad and RMF (Ni et al., 2022) on certain environments. The results shown for RMF are the average performance on the first 2 episodes, as taken from their repository, which did not contain results for more than 2 episodes and did not contain results for the Walker and Ant-Goal environments.

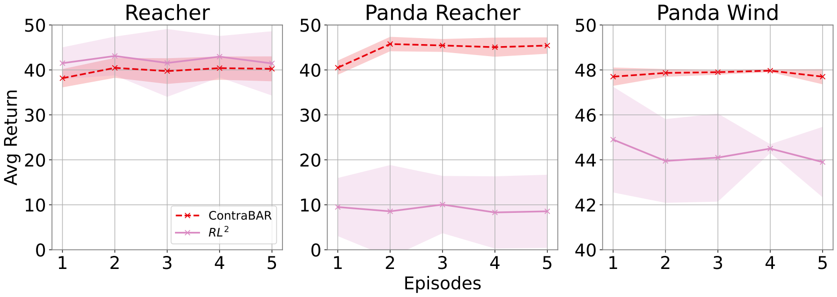

Figure 4: Average test performance on three different image-based environments, trained separately with different seeds per environment per method. The meta-trained policies are rolled out for 5 episodes to show how they adapt to the task. Values shown are averages across tasks (95% confidence intervals shaded). All methods were run with 5 seeds for each environment. We show competitive performance with RL^2, surpassing it in the Panda environments.

training two agents with ContraBAR on the Panda Reacher environment – one receives images modified by $\mathcal{T}$ and the other does not. We then evaluate each agent's performance on different color schemes, which are kept static for evaluation. The results as well as the environments can be seen in Figure 7. Note that while the belief may be robustified separately with augmentations, the policy must be robust to such changes as well. To do so, we used the data-regularized actor-critic method from Raileanu et al. (2021) where the policy π_θ and value function V_ϕ are regularized via two additional loss terms,

$$G_\pi = KL\left[\pi_\theta(a|s)|\pi_\theta(a|T(s))\right],$$
$$G_V = \left(V_\phi(s) - V_\phi(T(s))\right)^2,$$

where $T : \mathcal{S} \to \mathcal{S}$ randomly modifies the image.

We emphasize that domain randomization, as applied here, is not naturally compatible with variational belief inference methods. The reason is that when the loss targets *reconstruction* of the modified observation, the learned embedding cannot be trained to be *invariant* to the modification $\mathcal{T}$.

5.5. Offline ContraBAR

We show that as in VariBAD (Zintgraf et al., 2020), the disentanglement of belief and control allows us to reframe the algorithm within the context of offline meta RL, as was done in Dorfman et al. (2021). First, we use ContraBAR to learn a history embedding c_t from an offline dataset. Note that no specific change is required to our algorithm – we simply treat the offline dataset as the replay buffer for ContraBAR. Second, we perform *state relabeling* as described

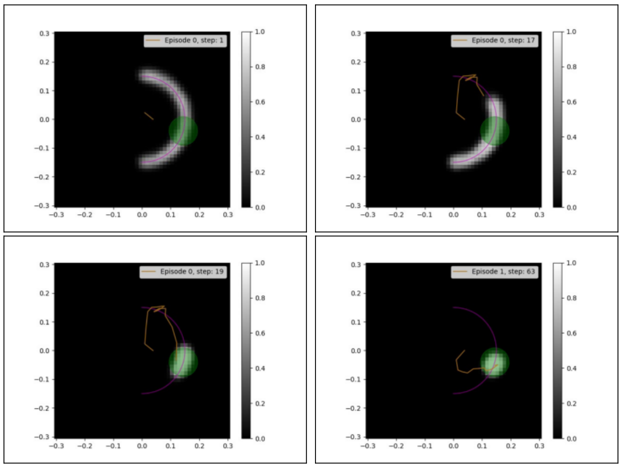

Figure 5: Belief visualization on the Panda Reacher environment. At each step, we visualise the predictions of the MLP-classifier for the entire map. Each frame shows an interesting milestone along the trajectory. From left to right along the rows: (1) Initially, the agent's belief over the location of the reward is uniform over the semi-circle, as in the prior. (2) As the agent travels along the semi-circle, the uncertainty regarding the location of the goal collapses in locations where the agent has visited and not found a goal. (3) Once the agent reaches the goal, all uncertainty collapses, even for locations the agent has yet to visit. (4) On the second episode, the agent follows the maintained belief and goes directly to the goal without exploration.

in Dorfman et al. (2021): for each trajectory τ_i of length T, i.e $(s_0^i, a_o^i, r_0^i, \ldots, s_T^i)$, we embed each partial t-length history h_t as c_t, and transform each s_t^i to $s_t^{+,i} = (s_t^i, c_t^i)$ as in the BAMDP formulation. We then learn a policy with SAC (Haarnoja et al., 2018) on the transformed dataset. We show competitive results with BOReL (Dorfman et al., 2021) in Figure 6. Unfortunately we were not able to find an offline adaptation of RMF to use as an additional baseline.

6. Conclusions

We proved that ContraBAR learns a representation that is a sufficient static of the history. Following on this, we presented what is to the best of our knowledge the first approximately Bayes-optimal CL meta RL algorithm. We demonstrated results competitive with previous approaches on several challenging state-input domains. Furthermore, by using contrastive learning we were able to scale meta-RL to image-based domains; We displayed results on par with RL^2 which was also able to scale to image inputs. Finally, we showed that our method is naturally amenable to domain randomization, which may be important for applications such as robotics.

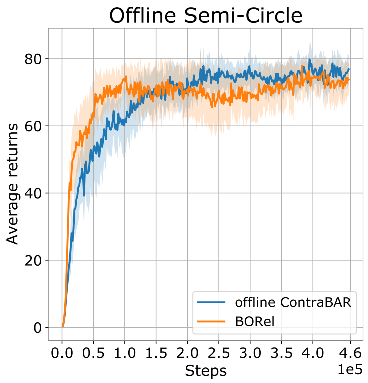

Figure 6: Results for the sparse Semi-Circle environment in the offline setting. Values shown are the average returns over two episodes, averaged over tasks. The shaded areas indicate 95% confidence intervals over different seeds of the method. ContraBAR and BOReL were both run with 10 seeds with reward relabeling.

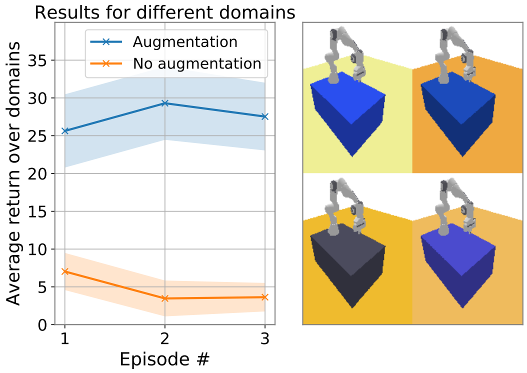

Figure 7: **Left:** Average test performance on three Panda Reacher environments with different color schemes from the training data. We show two meta-trained policies, one with augmentations and one without, each rolled out for 3 episodes to show how they adapt to the task. Values shown are averages across tasks (95% confidence intervals shaded). The agent trained with augmentations is invariant of the color schemes. **Right:** Top left image is the training environment; others are different color schemes for evaluation.

7. Acknowledgements

We thank Tom Jurgenson, Ev Zisselman, Orr Krupnik and Gal Avineri for useful discussions and feedback, and Luisa Zintgraf for invaluable help with reproducing the graphs from the VariBAD paper. This work received funding from the European Union (ERC, Bayes-RL, Project Number 101041250). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Bertsekas, D. P. *Dynamic programming and optimal control*, volume 1. Athena Scientific, 1995.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Clavera, I., Rothfuss, J., Schulman, J., Fujita, Y., Asfour, T., and Abbeel, P. Model-based reinforcement learning via meta-policy optimization. In *Conference on Robot Learning*, pp. 617–629, 2018.

Cover, T. M. and Thomas, J. A. Elements of information theory.

Dorfman, R., Shenfeld, I., and Tamar, A. Offline meta reinforcement learning–identifiability challenges and effective data collection strategies. *Advances in Neural Information Processing Systems*, 34, 2021.

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

Duff, M. O. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. PhD thesis, University of Massachusetts at Amherst, 2002.

Fakoor, R., Chaudhari, P., Soatto, S., and Smola, A. J. Meta-q-learning. *arXiv preprint arXiv:1910.00125*, 2019.

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

Fu, H., Tang, H., Hao, J., Chen, C., Feng, X., Li, D., and Liu, W. Towards effective context for meta-reinforcement learning: an approach based on contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7457–7465, 2021a.

Fu, L., Li, X., Wang, R., Zhang, Z., Wu, Y., He, X., and Zhou, B. Scala: Supervised contrastive learning for end-to-end automatic speech recognition. *arXiv preprint arXiv:2110.04187*, 2021b.

Gallouédec, Q., Cazin, N., Dellandréa, E., and Chen, L. panda-gym: Open-Source Goal-Conditioned Environments for Robotic Learning. *4th Robot Learning Workshop: Self-Supervised and Lifelong Learning at NeurIPS*, 2021.

Ghavamzadeh, M., Mannor, S., Pineau, J., and Tamar, A. Bayesian reinforcement learning: A survey. *arXiv preprint arXiv:1609.04436*, 2016.

Grant, E., Finn, C., Levine, S., Darrell, T., and Griffiths, T. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930*, 2018.

Guo, Z. D., Azar, M. G., Piot, B., Pires, B. A., and Munos, R. Neural predictive belief representations. *arXiv preprint arXiv:1811.06407*, 2018.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Han, T., Huang, H., Yang, Z., and Han, W. Supervised contrastive learning for accented speech recognition. *arXiv preprint arXiv:2107.00921*, 2021.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

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.

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

Laskin, M., Srinivas, A., and Abbeel, P. Curl: Contrastive unsupervised representations for reinforcement learning. *Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria, PMLR 119*, 2020. arXiv:2004.04136.

Laskin, M., Liu, H., Peng, X. B., Yarats, D., Rajeswaran, A., and Abbeel, P. Cic: Contrastive intrinsic control for unsupervised skill discovery. *arXiv preprint arXiv:2202.00161*, 2022.

Li, J., Vuong, Q., Liu, S., Liu, M., Ciosek, K., Christensen, H. I., and Su, H. Multi-task batch reinforcement learning with metric learning. *arXiv preprint arxiv:1909.11373*, 2020a.

Li, L., Yang, R., and Luo, D. Focal: Efficient fully-offline meta-reinforcement learning via distance metric learning and behavior regularization. *arXiv preprint arXiv:2010.01112*, 2020b.

Li, L., Huang, Y., Chen, M., Luo, S., Luo, D., and Huang, J. Provably improved context-based offline meta-rl with attention and contrastive learning. *arXiv preprint arXiv:2102.10774*, 2021.

Ni, T., Eysenbach, B., and Salakhutdinov, R. Recurrent model-free rl can be a strong baseline for many pomdps. In *International Conference on Machine Learning*, pp. 16691–16723. PMLR, 2022.

Nichol, A., Pfau, V., Hesse, C., Klimov, O., and Schulman, J. Gotta learn fast: A new benchmark for generalization in rl. *arXiv preprint arXiv:1804.03720*, 2018.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Ortega, P. A., Wang, J. X., Rowland, M., Genewein, T., Kurth-Nelson, Z., Pascanu, R., Heess, N., Veness, J., Pritzel, A., Sprechmann, P., et al. Meta-learning of sequential strategies. *arXiv preprint arXiv:1905.03030*, 2019.

Rabinovitz, C., Grupen, N., and Tamar, A. Unsupervised feature learning for manipulation with contrastive domain randomization. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10153–10159. IEEE, 2021.

Raileanu, R., Goldstein, M., Yarats, D., Kostrikov, I., and Fergus, R. Automatic data augmentation for generalization in reinforcement learning. *Advances in Neural Information Processing Systems*, 34:5402–5415, 2021.

Rakelly, K., Zhou, A., Quillen, D., Finn, C., and Levine, S. Efficient off-policy meta-reinforcement learning via probabilistic context variables. *arXiv preprint arXiv:1903.08254*, 2019.

Rothfuss, J., Lee, D., Clavera, I., Asfour, T., and Abbeel, P. Promp: Proximal meta-policy search. *arXiv preprint arXiv:1810.06784*, 2018.

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

Subramanian, J., Sinha, A., Seraj, R., and Mahajan, A. Approximate information state for approximate planning and reinforcement learning in partially observed systems. 2020.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Wang, B., Xu, S., Keutzer, K., Gao, Y., and Wu, B. Improving context-based meta-reinforcement learning with self-supervised trajectory contrastive learning. *arXiv preprint arXiv:2103.06386*, 2021.

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.

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. In *International Conference on Learning Representation (ICLR)*, 2020.

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. Theorem proofs

Theorem 4.3. *Let Assumption 4.2 hold. Let g^*, g^*_{AR}, f^* jointly minimize $\mathcal{L}_M(g, g_{AR}, f)$. Then the context latent representation $c_t = g^*_{AR}(z_{\leq t})$ satisfies conditions **P1, P2** and is therefore an information state.*

We begin our proof by presenting the causal model for the variant of CPC used by ContraBAR, shown in Figure 8.

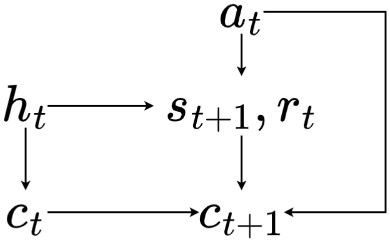

Figure 8: The causal model for c_t

From the causal model, we can infer that

$$P(c_{t+1}|c_t, s_{t+1}, r_t, a_t) =$$
$$P(c_{t+1}|c_t, s_{t+1}, r_t, a_t, h_t)$$

and $P(s_{t+1}, r_t|h_t, a_t) = P(s_{t+1}, r_t|h_t, a_t, c_t)$. We shall also assume that c_t is a deterministic function of h_t, and therefore $P(c_{t+1}|c_t, s_{t+1}, r_t, a_t, h_t) = P(c_{t+1}|s_{t+1}, r_t, a_t, h_t)$, and from the above, we have $P(c_{t+1}|s_{t+1}, r_t, a_t, h_t) = P(c_{t+1}|s_{t+1}, r_t, a_t, c_t)$.

We now prove a mutual information bound similar to that of Oord et al. (2018), we show that by optimizing the meta RL InfoNCE loss defined in Equation (1) we maximize the mutual information between c_t and s_{t+1}, r_t given a_t.

We begin with a lemma similar to that of Section 2.3 in Oord et al. (2018):

Definition A.1 (Possible sufficient statistic transition)**.** Let c_t be a function of h_t, i.e $c_t = \sigma_t(h_t)$. s_{t+1}, r_t, c_t, a_t is a possible sufficient statistic transition if h_t, a_t, r_t, s_{t+1} is a possible history as in Definition 4.1 and $c_t = \sigma_t(h_t)$.

Lemma A.2. *Let Assumption 4.2 and the loss in Equation* (1) *be jointly minimized by f, g, g_{AR}, then for any possible sufficient statistic transition s_{t+1}, r_t, c_t, a_t as in Definition A.1, where $c_t = g_{AR}(h_t)$, we have that*

$$f(s_{t+1}, r_t, c_t, a_t) \propto \frac{P(s_{t+1}, r_t|c_t, a_t)}{P(s_{t+1}, r_t|a_t)}$$

.

Proof. The loss in Eq. 1 is the categorical cross-entropy of classifying the positive example correctly, with $\frac{f}{\sum_B f}$ being the prediction of the model. We denote the j-th example in the batch B as s_j, r_j, where the subscript *does not* refer to time here. As in (Oord et al., 2018), the optimal probability for this loss is $P(d = i|B, c_t, a_t)$ (with $[d = i]$ indicating the i-th example in B is the positive example) and can be derived as follows:

$$\begin{aligned} P(d = i|B, c_t, a_t) &= \frac{P(s_i, r_i|c_t, a_t)\Pi_{l\neq i}P(s_l, r_l|a_t)}{\sum_{j=1}^{M} P(s_j, r_j|c_t, a_t)\Pi_{l\neq j}P(s_l, r_l|a_t)} \\ &= \frac{\frac{P(s_i, r_i|c_t, a_t)}{P(s_i, r_i|a_t)}}{\sum_{j=1}^{M} \frac{P(s_j, r_j|c_t, a_t)}{P(s_j, r_j|a_t)}}. \end{aligned} \tag{6}$$

□

Eq. 6 means that for any s_{t+1}, r_t, c_t, a_t that are part of a batch B in the data, we have that $f(s_{t+1}, r_t, c_t, a_t) \propto \frac{P(s_{t+1},r_t|c_t,a_t)}{P(s_{t+1},r_t|a_t)}$. From Assumption 4.2, for any sufficient statistic transition tuple s_{t+1}, r_t, c_t, a_t there exists a batch it is a part of.

Lemma A.3. *Let Assumption 4.2, and let the loss in Equation* (1) *be jointly minimized by* f, g, g_{AR}*. Then*

$$I(s_{t+1}, r_t; c_t|a_t) \geq \log(M-1) - \mathcal{L}_{opt}.$$

Proof. Given the optimal value shown in Lemma A.2 for $f(s_{t+1}, r_t, c_t, a_t)$, by inserting back into the loss we get:

$$\begin{aligned}
\mathcal{L}_{opt} &= -\mathbb{E}\log\left[\frac{\frac{P(s_{t+1},r_t|c_t,a_t)}{P(s_{t+1},r_t|a_t)}}{\frac{P(s_{t+1},r_t|c_t,a_t)}{P(s_{t+1},r_t|a_t)} + \sum_{(s',r')\in\{o_j^-\}_{j=1}^{M-1}} \frac{P(s',r'|c_t,a_t)}{P(s',r'|a_t)}}\right] \\
&= \mathbb{E}\log\left[1 + (\frac{P(s_{t+1},r_t|a_t)}{P(s_{t+1},r_t|c_t,a_t)} \sum_{(s',r')\in\{o_j^-\}_{j=1}^{M-1}} \frac{P(s',r'|c_t,a_t)}{P(s',r'|a_t)}\right] \\
&\approx \mathbb{E}\log\left[1 + (\frac{P(s_{t+1},r_t|a_t)}{P(s_{t+1},r_t|c_t,a_t)}(M-1)\cdot\mathbb{E}_{\mathcal{D}(s',r'|a_t)}\frac{P(s',r'|c_t,a_t)}{P(s',r'|a_t)}\right] \\
&= \mathbb{E}\log\left[1 + (\frac{P(s_{t+1},r_t|a_t)}{P(s_{t+1},r_t|c_t,a_t)}(M-1)\right] \\
&\geq \mathbb{E}\log\left[\frac{P(s_{t+1},r_t|a_t)}{P(s_{t+1},r_t|c_t,a_t)}(M-1)\right] \\
&= -I(s_{t+1}, r_t; c_t|a_t) + \log(M-1).
\end{aligned}$$

We therefore get that

$$I(s_{t+1}, r_t; c_t|a_t) \geq \log(M-1) - \mathcal{L}_{opt}.$$

We conclude that the objective maximizes the mutual information between c_t and s_{t+1}, r_t given a_t. □

Corollary A.4. *Let Assumption 4.2, and let the loss in Equation* (1) *be jointly minimized by* f, g, g_{AR}*, then* $I(c_t; s_{t+1}, r_t|a_t) = I(h_t; s_{t+1}, r_t|a_t)$ *where* $I(\cdot;\cdot)$ *denotes mutual information.*

Proof. Since s_{t+1}, r_t depend only on h_t (conditioned on a_t), and since c_t is a deterministic function of h_t, $I(s_{t+1}, r_t; c_t|a_t)$ cannot be greater than $I(s_{t+1}, r_t; h_t|a_t)$. From Lemma A.3 , we therefore have that

$$I(c_t; s_{t+1}, r_t|a_t) = I(h_t; s_{t+1}, r_t|a_t)$$

. □

Note that Corollary A.4 states that given the causal model above, c_t is maximally informative about s_{t+1}, r_t (conditioned on a_t). We use this result to prove a short lemma that will help us prove that that c_t is an information state.

Lemma A.5. *Let the assumptions of Corollary A.4 hold, then for every* a*,* $P(s_{t+1}, r_t|h_t, a_t) = P(s_{t+1}, r_t|c_t, a_t)$.

Proof. We start with a result similar to the data processing inequality. Consider $I(s_{t+1}, r_t; h_t, c_t|a_t)$. We have that

$$\begin{aligned}
&I(s_{t+1}, r_t; h_t, c_t|a_t) = \\
&I(s_{t+1}, r_t; c_t|h_t, a_t) + I(s_{t+1}, r_t; h_t|a_t),
\end{aligned} \tag{7}$$

and on the other hand,

$$\begin{aligned}
&I(s_{t+1}, r_t; h_t, c_t|a_t) = \\
&I(s_{t+1}, r_t; h_t|c_t, a_t) + I(s_{t+1}, r_t; c_t|a_t).
\end{aligned} \tag{8}$$

From the causal graph above, we have that $I(s_{t+1}, r_t; c_t|h_t, a_t) = 0$. Therefore, from Eq. (7) and (8) we have

$$\begin{aligned} I(s_{t+1}, r_t; h_t|a_t) &= I(s_{t+1}, r_t; c_t|a_t) + I(s_{t+1}, r_t; h_t|c_t, a_t) \\ &\geq I(s_{t+1}, r_t; c_t|a_t) \end{aligned}$$

with equality only if $I(s_{t+1}, r_t; h_t|c_t, a_t) = 0$, since the mutual information is positive. From Corollary A.4, we therefore must have $I(s_{t+1}, r_t; h_t|c_t, a_t) = 0$. This implies that s_{t+1}, r_t and h_t are independent conditioned on c_t, a_t (Cover & Thomas), and therefore

$$P(s_{t+1}, r_t|h_t, a_t) = P(s_{t+1}, r_t|c_t, a_t)$$

.

□

Proposition A.6. *Let Assumption 4.2, and let the loss in Equation* (1) *be jointly minimized by f, g, g_{AR}, then c_t satisfies **P1**, i.e.,* $\mathbb{E}[r_t|h_t, a_t] = \mathbb{E}[r_t|c_t, a_t]$.

Proof.

$$\begin{aligned} \mathbb{E}[r_t|h_t, a_t] &= \int r_t \int P(s_{t+1}, r_t|h_t, a_t) ds_{t+1} dr_t \\ &= \int r_t \int P(s_{t+1}, r_t|c_t, a_t) ds_{t+1} dr_t \\ &= \int r_t P(r_t|c_t, a_t) dr_t \\ &= \mathbb{E}[r_t|c_t, a_t]. \end{aligned}$$

□

Proposition A.7. *Let Assumption 4.2 and let the loss in Equation* (1) *be jointly minimized by f, g, g_{AR}, then c_t satisfies **P2**, i.e., $P(c_{t+1}|h_t) = P(c_{t+1}|c_t)$.*

Proof.

$$\begin{aligned} &P(c_{t+1}|h_t, a_t) \\ &= \int\int P(s_{t+1}, r_t|h_t, a_t) P(c_{t+1}|h_t, s_{t+1}, r_t, a_t) ds_{t+1} dr_t \\ &= \int\int P(s_{t+1}, r_t|c_t, a_t) P(c_{t+1}|h_t, c_t, s_{t+1}, r_t, a_t) ds_{t+1} dr_t \\ &= \int\int P(s_{t+1}, r_t|c_t, a_t) P(c_{t+1}|c_t, s_{t+1}, r_t, a_t) ds_{t+1} dr_t \\ &= P(c_{t+1}|c_t, a_t). \end{aligned}$$

where the second equality is due to lemma A.5 and the penultimate equality is due to c_{t+1} being a deterministic function of c_t, s_{t+1}, r_t and a_t □

We now provide the proofs for the setting described in Section 4.4, where there may be errors in the CPC learning, and the data does not necessarily satisfy Assumption 4.2.

We recapitulate that we consider an iterative policy improvement algorithm with a similarity constraint on consecutive policies, similar to the PPO algorithm we use in practice (Schulman et al., 2017). We shall bound the suboptimality of policy improvement, when data for training CPC is collected using the previous policy, denoted π_k. We will show optimal policy bounds when the information state is approximate, similar in spirit to Subramanian et al. (2020), but with additional technicalities. Under the setting above, we will bound the suboptimality in policy improvement in terms of an error in CPC training, which we denote ϵ.

In light of the bound from A.3, we assume the following:

Assumption 4.4. There exists an ϵ such that for every $t \leq T$, $I(s_{t+1}, r_t; c_t|a_t) \geq I(s_{t+1}, r_t; h_t|a_t) - \epsilon$, where the histories are distributed according to policy π_k.

We now define $P_\pi(h_t)$ as the probability of seeing a history under a policy π. For the sake of simplicity, for the subsequent section we will refer to $P_\pi(h_t)$ as $P(h_t)$. Furthermore, when the information state is approximate, we denote the information state generator $\hat{\sigma}_t$.

We begin with the following bound.

Proposition A.8. *Let Assumption 4.4 hold, then* $\mathbb{E}_{h_t \sim P(h_t)}\left[D_{KL}(P(s_{t+1}, r_t|h_t, a_t)||P(s_{t+1}, r_t|\hat{\sigma}_t(h_t), a_t)\right] \leq \epsilon$

Proof.

Proposition A.9. *Let Assumption 4.4 hold, then* $I(s_{t+1}, r_t; h_t|c_t, a_t) \leq \epsilon$

Proof. We start with a result similar to the data processing inequality. We have that

$$I(s_{t+1}, r_t; h_t, c_t|a_t) =$$
$$I(s_{t+1}, r_t; c_t|h_t, a_t) + I(s_{t+1}, r_t; h_t|a_t)$$

and,

$$I(s_{t+1}, r_t; h_t, c_t|a_t) =$$
$$I(s_{t+1}, r_t; h_t|c_t, a_t) + I(s_{t+1}, r_t; c_t|a_t)$$

From the causal graph we have that $I(s_{t+1}, r_t; c_t|h_t, a_t) = 0$, yielding

$$I(s_{t+1}, r_t; h_t|a_t) = I(s_{t+1}, r_t; h_t|c_t, a_t) + I(s_{t+1}, r_t; c_t|a_t) \Rightarrow$$
$$I(s_{t+1}, r_t; h_t|a_t) - I(s_{t+1}, r_t; c_t|a_t) = I(s_{t+1}, r_t; h_t|c_t, a_t)$$

Combined with 4.4 we get that $I(s_{t+1}, r_t; h_t|c_t, a_t) \leq \epsilon$ □

We note that from here on out everything is conditioned on a_t, and omit it to avoid overly cumbersome notation.

For ease of notation we define: $z = s_{t+1}, r_t$. We note that given a specific h_t, we have:

$$D_{KL}\left(P_{z|h_t}||P_{z|\hat{\sigma}_t(h_t)}\right) = \int_z P(z|h_t) \cdot \log\left(\frac{P(z|h_t)}{P(z|c_t)}\right)$$

Proposition A.10. $I(z; h_t|c_t) = \mathbb{E}_{h_t \sim P(h_t)}\left[D_{KL}\left(P_{z|h_t}||P_{z|\hat{\sigma}_t(h_t)}\right)\right]$

Proof.

$$\begin{aligned}
I(z;h_t|c_t) &= \mathbb{E}_{P_{\sigma_t(h_t)=c_t}}\left[D_{KL}\left(P_{z,h_t|c_t}||P_{z|c_t}\cdot P_{h_t|c_t}\right)\right] \\
&= \mathbb{E}_{P_{\sigma_t(h_t)=c_t}}\left[\int_{h_t}\int_z P(z,h_t|c_t)\,\log\left(\frac{P(z,h_t|c_t)}{P(z|c_t)\cdot P(h_t|c_t)}\right)\right] \\
&= \mathbb{E}_{P_{\sigma_t(h_t)=c_t}}\left[\int_{h_t}\int_z P(z,h_t|c_t)\,\log\left(\frac{P(z,h_t,c_t)\cdot P(c_t)}{P(z,c_t)\cdot P(h_t,c_t)}\right)\right] \\
&= \mathbb{E}_{P_{\sigma_t(h_t)=c_t}}\left[\int_{h_t}\int_z \frac{P(z,h_t,c_t)}{P(c_t)}\,\log\left(\frac{P(z|h_t)}{P(z|c_t)}\right)\right] \\
&= \int_{c_t}\int_{h_t}\int_z P(z,h_t,c_t)\,\log\left(\frac{P(z|h_t)}{P(z|c_t)}\right) \\
&= \int_{c_t}\int_{h_t}\int_z P(z|h_t)P(h_t,c_t)\,\log\left(\frac{P(z|h_t)}{P(z|c_t)}\right) \\
&= \int_{c_t}\int_{h_t} P(h_t,c_t)\int_z P(z|h_t)\,\log\left(\frac{P(z|h_t)}{P(z|c_t)}\right) \\
&= \int_{h_t} P(h_t)\int_{c_t}\delta_{c_t=\sigma_t(h_t)}\int_z P(z|h_t)\,\log\left(\frac{P(z|h_t)}{P(z|c_t)}\right) \\
&= \mathbb{E}_{h_t\sim P(h_t)}\left[D_{KL}(P(s_{t+1},r_t|h_t)||P(s_{t+1},r_t|\hat{\sigma}_t(h_t))\right]
\end{aligned}$$

□

We now complete the proof. Combining Proposition A.9 with Proposition A.10 we get that

$$\mathbb{E}_{h_t\sim P(h_t)}\left[D_{KL}\left(P_{s_{t+1},r_t|h_t}||P_{s_{t+1},r_t|\hat{\sigma}_t(h_t)}\right)\right]\le\epsilon$$

as required. □

Let $\pi_k(\hat{\sigma}_t(h_t))$ denote the policy at iteration k, and note that it is defined on the information state. At iteration $k+1$, we first collect data using π_k. We denote $P_{\pi_k}(h_t)$ the probability of observing a history in this data collection process. We then use CPC to learn an approximate information state. Let $D(h_t)=D_{KL}\left(P_{s_{t+1},r_t|h_t,a_t}||P_{s_{t+1},r_t|\hat{\sigma}_t(h_t),a_t}\right)$.

Proposition A.11. *Let Assumption 4.4 hold, then*

$$\sum_{h_t}P_{\pi_k}(h_t)D(h_t)\le\epsilon. \tag{9}$$

Proof. Assumption 4.4 holds, therefore the result is an immediate corollary from A.8 for every $t\in 0,1,\dots,T-1$. □

For some distance measure D, let $\Pi_\beta=\{\pi:D(\pi(h_t),\pi_k(\hat{\sigma}_t(h_t)))\le\beta\quad\forall h_t\}$ denote the set of policies that are β-similar to π_k.

We next define the optimal next policy π^*

$$\pi^*\in\underset{\pi\in\Pi_\beta}{\operatorname{argmax}}\,\mathbb{E}^\pi\left[\sum_{t=0}^{T-1}r(s_t,a_t)\right]. \tag{10}$$

Note that the value of this policy satisfies the following Bellman optimality equations:

$$\begin{aligned}
Q_t(h_t,a_t)&=r(h_t,a_t)+\mathbb{E}\left[V_{t+1}(h_{t+1})\right] \\
V_t(h_t)&=\max_{\pi:D(\pi(h_t),\pi_k(h_t))\le\beta}\sum_a\pi(a)Q_t(h_t,a),
\end{aligned} \tag{11}$$

for $t\le T$, and $V_T(h_T)=0$.

We now present our main result, where we consider an iterative policy improvement scheme based on the approximate information state of ContraBAR and provide policy improvement bounds.

Theorem 4.5. *Let Assumption 4.4 hold for some representation c_t. Consider the distance function between two distributions $D(P_1(x), P_2(x)) = \max_x |P_1(x)/P_2(x)|$. We let $\hat{r}(c_t, a_t) = \mathbb{E}[r_t|c_t, a_t]$ and $\hat{P}(c'|c_t, a_t) = \mathbb{E}[\mathbf{1}(c_{t+1} = c')|c_t, a_t]$ denote an approximate reward and transition kernel, respectively. Define the value functions*

$$\begin{aligned}\hat{Q}_t(c_t, a_t) &= \hat{r}(c_t, a_t) + \sum_{c_{t+1}} \hat{P}(c_{t+1}|c_t, a_t)\hat{V}_{t+1}(c_{t+1})\\ \hat{V}_t(c_t) &= \max_{\pi: D(\pi(c_t), \pi_k(c_t)) \leq \beta} \sum_a \pi(a)\hat{Q}_t(c_t, a),\end{aligned} \tag{4}$$

for $t \leq T$, and $\hat{V}_T(c_T) = 0$, and the approximate optimal policy

$$\hat{\pi}(c_t) \in \operatorname*{argmax}_{\pi: D(\pi, \pi_k(c_t)) \leq \beta} \sum_a \pi(a)\hat{Q}_t(c_t, a). \tag{5}$$

Let the optimal policy $\pi^*(h_t)$ be defined similarly, but with h_t replacing c_t in (4) *and* (5)*. Then we have that*

$$\mathbb{E}^{\pi^*}\left[\sum_{t=0}^{T-1} r(s_t, a_t)\right] - \mathbb{E}^{\hat{\pi}}\left[\sum_{t=0}^{T-1} r(s_t, a_t)\right] \leq \epsilon^{1/3} R_{max} T^2(\sqrt{2} + 4\beta^T).$$

Proof. Since Assumption 4.4 holds, Proposition A.11 does as well.

From the Markov inequality, we have $P_{\pi_k}(D(h_t) \geq n\epsilon) \leq \frac{\epsilon}{n\epsilon} = \frac{1}{n}$.

We now the define the "Good Set" $H_G = \{h_t : D(h_t) < n\epsilon\}$ and the "Bad Set" $H_B = \{h_t : D(h_t) \geq n\epsilon\}$.

Next, we define an auxiliary policy $\tilde{\pi}(h_t) = \begin{cases} \pi^*(h_t), & \text{if } h_t \in H_G \\ \text{worst behavior}, & \text{if } h_t \in H_B \end{cases}$.

We will assume that after observing $h_t \in H_B$, the policy performs as bad as possible for the rest of the episode.

Next, we bound the performance of $\tilde{\pi}$.

Proposition A.12. *We have that $\mathbb{E}^{\pi^*}\left[\sum_{t=0}^{T-1} r(s_t, a_t)\right] - \mathbb{E}^{\tilde{\pi}}\left[\sum_{t=0}^{T-1} r(s_t, a_t)\right] \leq 2T^2 R_{max}\beta^T/n$.*

Proof. We will denote by $r_t(h_t)$ the reward at the last state-action pair. That is, for $h_t = s_0, a_0, r_0, \ldots, s_{t-1}, a_{t-1}, r_{t-1}, s_t$ we set $r_t(h_t) = r_{t-1}$. We will denote $R(h_t)$ the sum of rewards, that is, $R(h_t) = \sum_{t'=0}^{t-1} r_{t'}$.

We also denote by $P_\pi(h_t)$ the probability of observing history h_t under policy π. Note that by definition $\sum_{t=0}^{T-1}\sum_{h_t} P_\pi(h_t) = 1$. Also, note that by the definition of the set Π_β, for any two policies $\pi_1, \pi_2 \in \Pi_\beta$ we have $P_{\pi_1}(h_t)/P_{\pi_2}(h_t) \leq \beta^t$.

We now claim that

$$\mathbb{E}^{\tilde{\pi}}\left[\sum_{t=0}^{T-1} r_t\right] \geq \mathbb{E}^{\pi^*}\left[\sum_{t=0}^{T-1} r_t\right] - 2T^2 R_{max}\beta^T/n.$$

We first estimate the probability that policy $\tilde{\pi}$ encounters a history in H_B. Consider some $t \in 0, \ldots, T-1$. We have that under P_{π_k}, with probability at most $1/n$, $h_t \in H_B$. Under $P_{\tilde{\pi}}$, with probability at most β^t/n, $h_t \in H_B$. From the union bound, with probability at most $T\beta^T/n$ the policy visits at least one history in H_B.

Let $\bar{H}_B$ denote the set of T-length histories that visit a history in H_B, and let $\bar{H}_G$ be its complement set.

Now, note that

$$\begin{aligned}
\mathbb{E}^{\tilde{\pi}}\left[\sum_{t=0}^{T-1} r_t\right] &= \sum_{h_T} P_{\tilde{\pi}}(h_T)R(h_T) \\
&= \sum_{h_T\in\bar{H}_G} P_{\tilde{\pi}}(h_T)R(h_T) + \sum_{h_T\in\bar{H}_B} P_{\tilde{\pi}}(h_T)R(h_T) \\
&= \sum_{h_T\in\bar{H}_G} P_{\pi^*}(h_T)R(h_T) + \sum_{h_T\in\bar{H}_B} P_{\tilde{\pi}}(h_T)R(h_T) \\
&\geq \sum_{h_T\in\bar{H}_G} P_{\pi^*}(h_T)R(h_T) + (T\beta^T/n)T(-R_{max}) \\
&= \sum_{h_T} P_{\pi^*}(h_T)R(h_T) - \sum_{h_T\in\bar{H}_B} P_{\pi^*}(h_T)R(h_T) + (T\beta^T/n)T(-R_{max}) \\
&\geq \mathbb{E}^{\pi^*}\left[\sum_{t=0}^{T-1} r_t\right] - 2T^2R_{max}\beta^T/n
\end{aligned} \tag{12}$$

The third equality is from the definition of π^*. The fourth inequality relies on the reward function being bounded, i.e $R(h_T) \geq T(-R_{max})$. This alongside the fact that $\sum_{h_T\in\bar{H}_B} P_{\tilde{\pi}}(h_T) \geq (TB^T/n)$ gives us the inequality. Note that the last inequality follows from the definition of π_*, wherein the probability of visiting at least one history in H_B is the same for π_* and $\tilde{\pi}$. □

Next, we note that using Pinsker's inequality, we have $d_{TV}(P_{s_{t+1},r_t|h_t,a_t}, P_{s_{t+1},r_t|c_t,a_t}) \leq \sqrt{2d_{KL}(P_{s_{t+1},r_t|h_t,a_t}, P_{s_{t+1},r_t|c_t,a_t})}$, and that

$$\begin{aligned}
|\mathbb{E}[r_t|h_t,a_t] - \mathbb{E}[r_t|c_t,a_t]| &\leq R_{max}d_{TV}(P_{s_{t+1},r_t|h_t,a_t}, P_{s_{t+1},r_t|c_t,a_t}) \\
|\mathbb{E}[V_{t+1}|h_t,a_t] - \mathbb{E}[V_{t+1}|c_t,a_t]| &\leq R_{max}(T-t)d_{TV}(P_{s_{t+1},r_t|h_t,a_t}, P_{s_{t+1},r_t|c_t,a_t})
\end{aligned}$$

We next prove the following result.

Proposition A.13. *We have that*

$$\begin{aligned}
\hat{Q}_t(\hat{\sigma}_t(h_t), a) &\geq Q^{\tilde{\pi}}(h_t, a) - \alpha_t, \\
\hat{V}_t(\hat{\sigma}_t(h_t)) &\geq V^{\tilde{\pi}}(h_t) - \alpha_t,
\end{aligned} \tag{13}$$

where α_t satisfies the following recursion: $\alpha_T = 0$, and $\alpha_t = \sqrt{2n\epsilon}R_{max}(T-t+1) + \alpha_{t+1}$.

Proof. We prove by backward induction. The argument holds for T by definition. Assume that Equation (13) holds at time $t+1$, and consider time t. If $h_t \in H_B$, then by definition $\hat{Q}_t(\hat{\sigma}_t(h_t), a) \geq Q^{\tilde{\pi}}(h_t, a)$, since $\tilde{\pi}$ will take the worst possible actions after observing h_t. Otherwise, $h_t \in H_G$ and we have

$$\begin{aligned}
&Q^{\tilde{\pi}}(h_t, a) - \hat{Q}_t(\hat{\sigma}_t(h_t), a) \\
=&\mathbb{E}[r_t|h_t,a] + \mathbb{E}\left[V_{t+1}^{\tilde{\pi}}(h_{t+1})|h_t,a\right] - \hat{r}(\hat{\sigma}_t(h_t),a) - \sum_{c_{t+1}} \hat{P}(c_{t+1}|\hat{\sigma}_t(h_t),a)\hat{V}_{t+1}(c_{t+1}) \\
=&\mathbb{E}[r_t|h_t,a] - \mathbb{E}[r_t|c_t,a_t] \\
&+ \mathbb{E}\left[V_{t+1}^{\tilde{\pi}}(h_{t+1})|h_t,a\right] - \mathbb{E}\left[\hat{V}_{t+1}(\hat{\sigma}_{t+1}(h_{t+1}))|h_t,a\right] \\
&+ \mathbb{E}\left[\hat{V}_{t+1}(\hat{\sigma}_{t+1}(h_{t+1}))|h_t,a\right] - \sum_{c_{t+1}} \hat{P}(c_{t+1}|\hat{\sigma}_t(h_t),a)\hat{V}_{t+1}(c_{t+1}) \\
\leq&\sqrt{2n\epsilon}R_{max} + \alpha_{t+1} + \sqrt{2n\epsilon}R_{max}(T-t).
\end{aligned}$$

We note that for $h_t \in H_G$, $D(h_t) \leq n\epsilon$, yielding the d_{TV} bounds.

For the second part, If $h_t \in H_B$, then by definition $\hat{V}_t(\hat{\sigma}_t(h_t)) \geq V^{\tilde{\pi}}(h_t)$. Otherwise, $h_t \in H_G$ and we have

$$\begin{aligned}
V_t^{\tilde{\pi}}(h_t) - \hat{V}_t(\hat{\sigma}_t(h_t)) &= \max_{\pi: D(\pi(h_t), \pi_k(h_t)) \leq \beta} \sum_a \pi(a) Q_t^{\tilde{\pi}}(h_t, a) - \max_{\pi: D(\pi(h_t), \pi_k(h_t)) \leq \beta} \sum_a \pi(a) \hat{Q}_t(\hat{\sigma}_t(h_t), a) \\
&\leq \max_{\pi: D(\pi(h_t), \pi_k(h_t)) \leq \beta} \sum_a \pi(a) (\hat{Q}_t(\hat{\sigma}_t(h_t), a) + \alpha_t) - \max_{\pi: D(\pi(h_t), \pi_k(h_t)) \leq \beta} \sum_a \pi(a) \hat{Q}_t(\hat{\sigma}_t(h_t), a) \\
&= \alpha_t.
\end{aligned}$$

□

We next define another auxiliary policy $\tilde{\hat{\pi}}(h_t) = \begin{cases} \hat{\pi}(h_t), & \text{if } h_t \in H_G \\ \text{optimal behavior}, & \text{if } h_t \in H_B \end{cases}$. We will assume that after observing $h_t \in H_B$, the policy perform optimally for the rest of the episode. Therefore, $V_t^{\tilde{\hat{\pi}}}(h_t \in H_B) = V_t(h_t)$. We have the following results, analogous to Propositions A.12 and A.13.

Proposition A.14. *We have that* $\mathbb{E}^{\tilde{\hat{\pi}}}\left[\sum_{t=0}^{T-1} r(s_t, a_t)\right] - \mathbb{E}^{\hat{\pi}}\left[\sum_{t=0}^{T-1} r(s_t, a_t)\right] \leq 2T^2 R_{max} \beta^T / n.$

Proof. Analogous to the proof of Proposition A.12. □

Proposition A.15. *We have that*

$$\begin{aligned}
\hat{Q}_t(\hat{\sigma}_t(h_t), a) &\leq Q^{\tilde{\hat{\pi}}}(h_t, a) + \alpha_t, \\
\hat{V}_t(\hat{\sigma}_t(h_t)) &\leq V^{\tilde{\hat{\pi}}}(h_t) + \alpha_t,
\end{aligned} \tag{14}$$

where α_t satisfies the following recursion: $\alpha_T = 0$, and $\alpha_t = \sqrt{2n\epsilon} R_{max}(T - t + 1) + \alpha_{t+1}$.

Proof. Similarly to the proof of Proposition A.13. The argument hold for T by definition. Assume that Equation (14) holds at time $t+1$, and consider time t. If $h_t \in H_B$, then by definition $\hat{Q}_t(\hat{\sigma}_t(h_t), a) \leq Q^{\tilde{\hat{\pi}}}(h_t, a)$, since $\tilde{\hat{\pi}}$ will take the best possible actions after observing h_t. Otherwise, $h_t \in H_G$ and we have

$$\begin{aligned}
&\hat{Q}_t(\hat{\sigma}_t(h_t), a) - Q^{\tilde{\hat{\pi}}}(h_t, a) \\
=&\hat{r}(\hat{\sigma}_t(h_t), a) + \sum_{c_{t+1}} \hat{P}(c_{t+1}|\hat{\sigma}_t(h_t), a) \hat{V}_{t+1}(c_{t+1}) - \mathbb{E}[r_t|h_t, a] - \mathbb{E}\left[V_{t+1}^{\tilde{\hat{\pi}}}(h_{t+1})|h_t, a\right] \\
=&\mathbb{E}[r_t|\hat{\sigma}_t(h_t), a_t] - \mathbb{E}[r_t|h_t, a] \\
&+ \mathbb{E}\left[\hat{V}_{t+1}(\hat{\sigma}_{t+1}(h_{t+1}))|h_t, a\right] - \mathbb{E}\left[V_{t+1}^{\tilde{\hat{\pi}}}(h_{t+1})|h_t, a\right] \\
&+ \sum_{c_{t+1}} \hat{P}(c_{t+1}|\hat{\sigma}_t(h_t), a) \hat{V}_{t+1}(c_{t+1}) - \mathbb{E}\left[\hat{V}_{t+1}(\hat{\sigma}_{t+1}(h_{t+1}))|h_t, a\right] \\
\leq&\sqrt{2n\epsilon} R_{max} + \alpha_{t+1} + \sqrt{2n\epsilon} R_{max}(T - t).
\end{aligned}$$

For the second part, If $h_t \in H_B$, then by definition $\hat{V}_t(\hat{\sigma}_t(h_t)) \leq V^{\tilde{\hat{\pi}}}(h_t)$. Otherwise, $h_t \in H_G$ and we have

$$\begin{aligned}
\hat{V}_t(\hat{\sigma}_t(h_t)) - V_t^{\tilde{\hat{\pi}}}(h_t) &= \max_{\pi: D(\pi(h_t), \pi_k(h_t)) \leq \beta} \sum_a \pi(a) \hat{Q}_t(\hat{\sigma}_t(h_t), a) - \max_{\pi: D(\pi(h_t), \pi_k(h_t)) \leq \beta} \sum_a \pi(a) Q_t^{\tilde{\hat{\pi}}}(h_t, a) \\
&\leq \max_{\pi: D(\pi(h_t), \pi_k(h_t)) \leq \beta} \sum_a \pi(a) (Q_t^{\tilde{\hat{\pi}}}(h_t, a) + \alpha_t) - \max_{\pi: D(\pi(h_t), \pi_k(h_t)) \leq \beta} \sum_a \pi(a) Q_t^{\tilde{\hat{\pi}}}(h_t, a) \\
&= \alpha_t.
\end{aligned}$$

□

Corollary A.16. *We have*

$$\begin{aligned}
&\mathbb{E}^{\pi^*}\left[\sum_{t=0}^{T-1} r(s_t,a_t)\right] - \mathbb{E}^{\hat{\pi}}\left[\sum_{t=0}^{T-1} r(s_t,a_t)\right] \leq \mathbb{E}^{\tilde{\pi}}\left[\sum_{t=0}^{T-1} r(s_t,a_t)\right] - \mathbb{E}^{\hat{\pi}}\left[\sum_{t=0}^{T-1} r(s_t,a_t)\right] + 2T^2 R_{max}\beta^T/n \\
&= \sum_{h_0} P(h_0)\left(V_0^{\tilde{\pi}}(h_0) - V_0^{\hat{\pi}}(h_0)\right) + 2T^2 R_{max}\beta^T/n \\
&\leq \sum_{h_0} P(h_0)\Big(V_0^{\tilde{\pi}}(h_0) - \hat{V}_0(\hat{\sigma}_0(h_0)) + \hat{V}_0(\hat{\sigma}_0(h_0)) - V_0^{\hat{\pi}}(h_0)\Big) + 2T^2 R_{max}\beta^T/n \\
&\leq \sum_{h_0} P(h_0)\Big(V_0^{\tilde{\pi}}(h_0) - \hat{V}_0(\hat{\sigma}_0(h_0)) + \hat{V}_0(\hat{\sigma}_0(h_0)) - V_0^{\tilde{\tilde{\pi}}}(h_0)\Big) + 4T^2 R_{max}\beta^T/n \\
&\leq 2\alpha_0 + 4T^2 R_{max}\beta^T/n
\end{aligned}$$

We note that the first inequality follows from Proposition A.12. The second equality stems from the fact that $P(h_0) = P(s_0)$, which is not affected by the choice of policy. The fourth transition follows from the addition and subtraction of $V_0^{\tilde{\tilde{\pi}}}(h_0)$ and the use of Proposition A.14. The final inequality follows from Propositions A.13 and A.15.

Let us bound α_0. By the recursion $\alpha_t = \sqrt{2n\epsilon}R_{max}(T-t+1) + \alpha_{t+1}$ we have that $\alpha_0 = \frac{T^2}{2}\sqrt{2n\epsilon}R_{max}$. Setting $n = \epsilon^{-1/3}$ we obtain the desired result:

$$\begin{aligned}
\mathbb{E}^{\pi^*}\left[\sum_{t=0}^{T-1} r(s_t,a_t)\right] - \mathbb{E}^{\hat{\pi}}\left[\sum_{t=0}^{T-1} r(s_t,a_t)\right] &\leq T^2\sqrt{2}\epsilon^{1/3}R_{max} + 4T^2 R_{max}\beta^T\epsilon^{1/3} \\
&= \epsilon^{1/3}R_{max}T^2(\sqrt{2} + 4\beta^T).
\end{aligned}$$

□

B. Environments

Reacher Image: In this environment, a two-link planar robot needs to reach an unknown goal as in (Dorfman et al., 2021), except that the goal is randomly chosen along a horizontal section of 0.48. For each task, the agent receives a reward of +1 if it is within a small radius $r = 0.05$ of the goal, and 0 otherwise.

$$r_t = \begin{cases} 1, & \text{if } \|x_t - x_{goal}\|_2 \leq 0.05 \\ 0, & \text{otherwise} \end{cases}$$

where x_t is the location of the robot's end effector. The agent observes single-channel images of size 64 × 64 of the environment]. The horizon is set to 50 and we aggregate $k = 2$ consecutive episodes to form a trajectory of length 100.

Panda Reacher: A Franka Panda robot tasked with placing the end effector at a goal on a 2d semi-circle of radius 0.15 with fixed $z = 0.15/2$ in 3d-space. The task is adapted from the Reacher task in Panda Gym (Gallouédec et al., 2021), with the goal occluded. For each task, the agent receives a reward of +1 if it is within a small radius $r = 0.05$ of the goal, and 0 otherwise.

$$r_t = \begin{cases} 1, & \text{if } \|x_t - x_{goal}\|_2 \leq 0.05 \\ 0, & \text{otherwise} \end{cases}$$

where x_t is the current location of the end effector. The action space is 3-dimensional and bounded $[-1, 1]^3$. The agent observes a 3-channel image of size 84×84 of the environment. We set the horizon to 50 and aggregate $k = 3$ consecutive episodes to form a trajectory of length 150.

Panda Wind: This environment is identical to Panda Reacher, except that the goal is fixed and for each task the agent experiences different wind with shifts the transition function, such that for an MDP $\mathcal{M}$ the transition function becomes

$$s_{t+1} = s_t + a_t + w_{\mathcal{M}}$$

where $w_{\mathcal{M}}$ is task specific and drawn randomly from a circle of radius 0.1. To get to the goal and stay there, the agent must learn to quickly adapt in a way that cancels the effect of the wind.

C. Implementation Details

In this section we outline our training process and implementation details, exact hyperparameters can be found in our code at `https://github.com/ec2604/ContraBAR`

The CPC component termed g_{AR} consists of a recurrent encoder, which at time step t takes as input the tuple (a_t, r_{t+1}, s_{t+1}). The state, reward and action are passed each through a different fc layer (or a cnn feature-extractor for the states in image-based inputs). Our CPC projection head takes in (c^a_t, z_{t+k}) and passes it through one hidden layer of half the input size, with an ELU activation function.

D. Architecture Details

In this section we detail practical considerations regarding the CPC architecture.

In Section 4.5 we described situations where c_t does not need to encode belief regarding the task in order to distinguish between positive and negative observations. This is detrimental to learning a sound sufficient statistic as we would like c_t to encode information regarding the reward and transition functions, as they are what set apart each task. In order to prevent this "shortcut" from being used, we can perform hard negative mining. We do this by using negative observations that cannot be distinguished from the positive observation without belief regarding the transition and reward functions. In the case where only the reward functions vary, we can do this by taking the state and action of the positive observation and sampling a new reward function. We then calculate the respective reward and embed it as a negative observation alongside the original state and action. By having the positive and negative observations share the same state and action, we ensure that c_t must be informative regarding the reward function in order to distinguish between positive and negative observations. We note that in this modified setup we use s_t as the initial hidden state for the action-gru and include the original c_t as input to the CPC projection head. This ensures that the gradient of the loss with respect to the action-gru does not affect c_t, which should encode information regarding the reward function. For the case where the environments only vary in reward functions, we propose a simpler solution which is to omit the action-gru, as the future actions except for a_{t+k-1} do not affect r_{t+k}. We can simply use (c_t, z_{t+k}) as input to the CPC projection head – we note that in this case z_{t+k} is an embedding of the reward, state *and action*. We found in practice that this simplification also worked well for the environments we used where the transitions varied.

The modified architecture where the action-gru is omitted can be seen in Figure 9. In Figure 10 we demonstrate on the Ant-Goal environment that omitting the action-gru and reward-relabeling *with* the action-gru yield similar results. Finally, we note that hard-negative mining can be done for varying transitions by sampling a random transition from the prior and simulating the transition to some s_{t+k} given $s_t, a_t, \ldots, a_{t+k}$.

E. Image-based inputs are computationally restrictive for VariBAD

To understand the computational restriction in VariBAD (Zintgraf et al., 2020), we look to the formulation of the VAE objective. For every timestep t, the past trajectory $\tau_{:t}$ is encoded to infer the posterior $q(m|\tau_{:t})$, and used by the decoder to reconstruct *the entire* trajectory including the future. In our analysis we restrict ourselves to the memory required for reconstruction of the reward trajectory, in an image-based domain, under the following assumptions: (1) images of dimension $d \times d \times 3$ are embedded to a representation of size 32 via 3 convolutions with 32 channels each and kernels of size, with strides $2, 2, 1$ respectively. (2) actions are of size 2 and embedded with a linear layer of size 16 (3) The trajectory is of length 120, which is average for the domains in meta RL. We draw attention to the fact that reward decoder in VariBAD receives s_t, s_{t+1} as input, requiring us to take into consideration the memory required for embedding the image trajectory. On top of this, we also consider three times the size of the parameters of the image encoder (parameters, gradients and gradient moments). We present the memory consumption as a function of the image dimensions in Figure 11. We note that in practice we often wish to decode multiple trajectories at once, and we also need to take into account encoder portion of the model as well as its gradients.

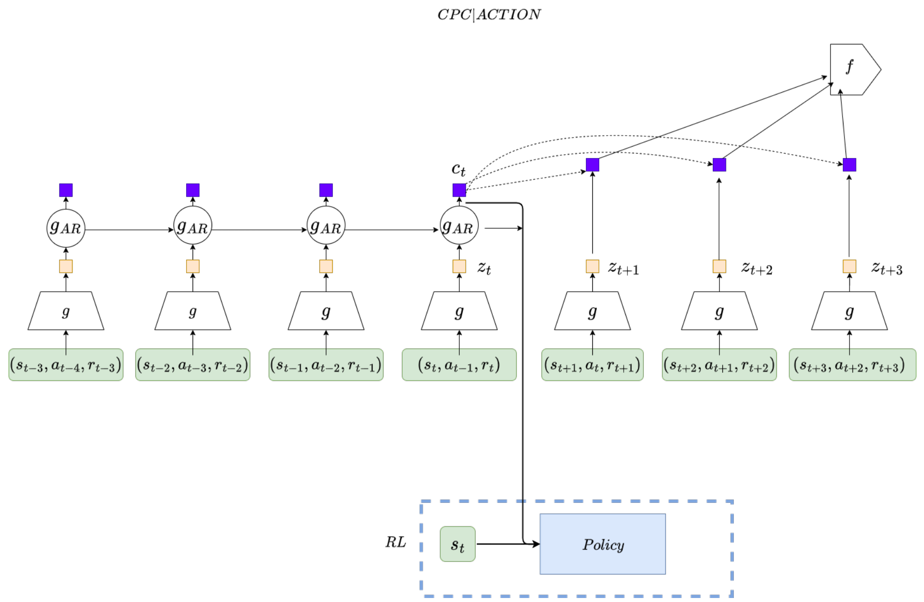
CPC|ACTION
f
c_t
g_{AR}
g_{AR}
g_{AR}
g_{AR}
z_t
z_{t+1}
z_{t+2}
z_{t+3}
g
g
g
g
g
g
g
$(s_{t-3}, a_{t-4}, r_{t-3})$
$(s_{t-2}, a_{t-3}, r_{t-2})$
$(s_{t-1}, a_{t-2}, r_{t-1})$
(s_t, a_{t-1}, r_t)
(s_{t+1}, a_t, r_{t+1})
$(s_{t+2}, a_{t+1}, r_{t+2})$
$(s_{t+3}, a_{t+2}, r_{t+3})$
RL
s_t
Policy

Figure 9: ContraBAR architecture where the action-gru is omitted. The history encoding is the same, future observations however, do not include the action.

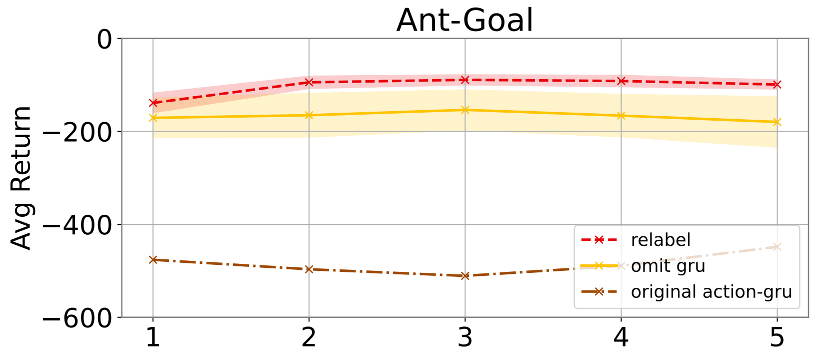

Figure 10: The relabel and omit-gru architectures are run on the Ant-goal environment, with 10 seeds (evaluated over 10 environment samples each). The meta-trained policies are rolled out for 5 episodes to show how they adapt to the task. Values shown are averages across tasks (95% confidence intervals shaded). The results for both methods are similar, with slightly better results when the action-gru is omitted. We additionally show a single seed of the original action-gru architecture, evaluated on 10 environment samples – we see that the sample mean is out of the confidence interval for both the relabel and omit-gru architectures.

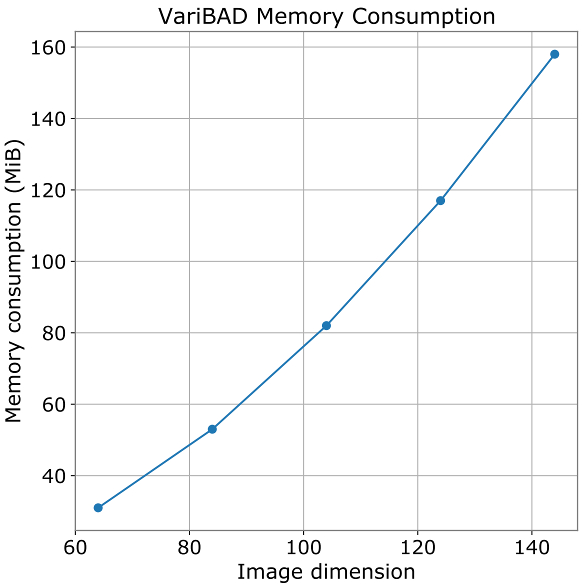

Figure 11: VariBAD memory consumption for decoding the rewards for image-based input, as a function of the image dimension given a trajectory length of 200