LEARNING TASK INFORMED ABSTRACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Current model-based reinforcement learning methods struggle when operating from complex visual scenes due to their inability to prioritize task-relevant features. To mitigate this problem, we propose learning **Task Informed Abstractions** (TIA) that separates reward-correlated visual features from background distractions. For learning TIA, we introduce the formalism of Task Informed MDP (TiMDP), which is realized by training two models that learn visual features via cooperative reconstruction, but one model is adversarially dissociated from the reward signal. Empirical evaluation shows that TIA leads to significant performance gains over state-of-the-art methods on many visual control tasks where natural and unconstrained visual distractions pose a formidable challenge.

1 INTRODUCTION

Consider results of a simple experiment reported in Figure 1. We train state-of-the-art model-based reinforcement learning agents (Hafner et al., 2020b) operating from visual inputs to solve two versions of the *Cheetah Run* task (Tassa et al., 2018): one with a simple and the other with a visually complex background (Zhang et al., 2020b). For each version we train three model variants that only differ in in the capacity of their world model, which contain $0.5\times$ (small), $1\times$ (medium) and $2\times$ (large) parameters of the original implementation. Not surprisingly, the performance with the simple background is only marginally affected by model capacity, which shows that even the smallest model is sufficient for learning the features necessary for solving the task. Performance is much worse with the complex background, but it increases monotonically with the model size. Given that the task-relevant information (i.e., joint information of the *cheetah*) is the same in both, the performance improvements with model size indicates that the increase in representational capacity is used to encode the complex background. The background conveys no information about the task and therefore interferes with the learning of task-relevant information by consuming model capacity.

There are two main components to a model-based learner: (i) a forward dynamics model, and (ii) a reward and value predictor. The policy performance critically depends on the prediction accuracy of this model, which is intimately tied to the quality of the learned feature space. Similar to the version of *Cheetah run* with the complex background, the real world are full of irrelevant content that clutters the observation. Therefore to learn directly in the real-world, a compact representation that only captures task-relevant features could make downstream policy learning much easier.

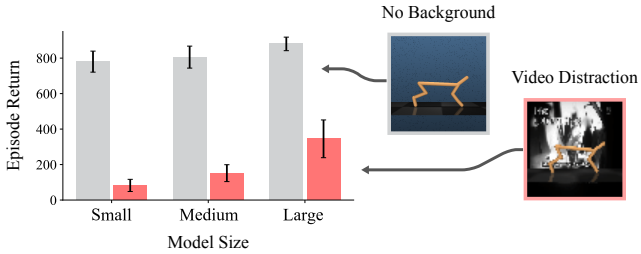


Figure 1: Comparison of the performance of a state-of-the-art model-based RL algorithm, *Dreamer*, on two versions of the *Cheetah Run* with vs. without visual distraction. Performance is reported for three models of different sizes ($2\times, 1\times, 0.5\times$ of original Dreamer). Results show that even the smallest model has sufficient capacity to capture task-relevant features when observations are distractor-free (gray), but when the scene is complex (red), task-irrelevant features consume most of the model capacity. Error bars indicate one standard deviation.

So far, a popular choice for feature learning is to reconstruct the raw observations (Kingma & Welling, 2014; Kingma et al., 2014; Watter et al., 2015; Hafner et al., 2020b). Often these features are encouraged

to be *disentagled* (Bengio, 2013; Higgins et al., 2016; 2017) to identify distinct factors of variation. Since disentanglement simply re-formats the input space, the disentangled feature space would still contain irrelevant information and does not address the core issue of learning task-relevant features. Because the model-based agent also predicts rewards from the feature space, one might expect the combination of disentanglement and reward prediction sufficient to incentivize learning of task-relevant features. However, rewards provide insufficient supervision for feature learning (Yarats et al., 2019). For instance, just knowing the center of mass of a humanoid moving forward is sufficient to predict the reward, whereas it would require the full pose to come up with the optimal action. In a nutshell *reconstruction captures too much information*, whereas *reward-prediction captures too little*. Several works attempt to combine these two training signals (Hafner et al., 2020b; Oh et al., 2017) but still struggle to learn in complex visual scenarios. Since the goal of the agent is to maximize the expected return, predicting the value function instead of one-step reward may aid in learning all the relevant information (Silver et al., 2016; Oh et al., 2017). However, because the value function is learned simultaneously with the model, it is not stationary and may not provide a stable training signal.

These challenges inspired several works to investigate feature learning methods that neither rely on reconstruction nor solely depend on rewards. One line of work biases the learned features to only capture controllable parts of the environment using an inverse model that predicts actions from a pair of states (Agrawal et al., 2015; Jayaraman & Grauman, 2015; Agrawal et al., 2016; Pathak et al., 2017), or using metrics such as empowerment (Klyubin et al., 2005; Gregor et al., 2016). To understand their shortcoming, consider the scenario of the arm pushing an object. Here both the arm and the object are controllable. While it is easy to capture the part that is directly controllable (e.g., the arm), capturing all controllable features (i.e., arm and the object) without imposing a reconstruction loss is non-trivial. Another idea that has shown promise is the bisimulation metric (Ferns et al., 2011; Zhang et al., 2020b). Because supervision in bisimulation comes solely from rewards, it is subject to the same insufficiency mentioned earlier. Another possibility is to use contrastive learning (Chen et al., 2020; Oord et al., 2018), but without additional constraints, these methods do not distinguish between relevant and irrelevant features.

In this work, we revisit feature learning by using reconstruction and the reward but propose to explicitly "explain away" irrelevant features by constructing a co-operative two-player game between two models. These models, dubbed as primary and secondary, learn task-relevant (s_t^+) and irrelevant features (s_t^-) of the observation (o_t) respectively. Similar to prior work, we force the primary model to learn task-relevant features (s_t^+) by predicting the reward. But unlike past work, we also force the secondary model to learn task-irrelevant features (s_t^-) via adversarial dissociation with the reward signal. However, both models cooperate to reconstruct o_t by maximizing $p(o_t|s_t^+, s_t^-)$.

Our method models a Markov decision process (MDP) of a specific factored structure, which we call **Task Informed MDP (TiMDP)** (see Figure 2b). It is worth noting that TiMDP is structurally similar to the *relaxed block MDP* (Zhang et al., 2020a) formulation in partitioning the state-space into two separate components. However, Zhang et al. 2020a neither proposes a practical method for segregating relevant information nor does it provide any experimental validation in the context of learning from complex visual inputs. We evaluate our method on a custom ManyWorld environment, a suite of control tasks that specifically test the robustness of learning to visual distractions (Zhang et al., 2020b) and ATARI games. The results convincingly demonstrate that our method, which we call **T**ask **I**nformed **A**bstractions (TIA), successfully learns relevant features and outperforms existing state-of-the-art methods.

2 PRELIMINARIES

A Markov Decision Process is represented as the tuple $\langle \mathcal{S}, \mathcal{O}, \mathcal{A}, T, r, \gamma, \rho_0 \rangle$ where $\mathcal{O}$ is a high-dimensional observation space. $\mathcal{A}$ is the space of actions. $\mathcal{S}$ is the state space. ρ_0 is the initial state distribution. $r : \mathcal{S} \mapsto \mathbb{R}$ is the scalar reward. The goal of RL is to learn a policy $\pi^*(a|\, s)$ that maximizes cumulative reward $\mathcal{J}_\pi = \arg\max_\pi \mathbb{E} \sum_t \gamma^{t-1} r_t$ discounted by γ.

2.1 MODEL-BASED RL

Our primary contribution is in learning a feature space for forward dynamics and it is agnostic to the specific choice of the model-based algorithm. We choose to build upon the state-of-the-art method known as Dreamer (Hafner et al., 2020b). The main components of this model are:

$$\begin{aligned} &\text{Representation model:} && p_\theta(s_t \mid o_t, s_{t-1}, a_{t-1}) \\ &\text{Observation model:} && q_\theta(o_t \mid s_t) \\ &\text{Transition model:} && q_\theta(s_t \mid s_{t-1}, a_{t-1}) \\ &\text{Reward model:} && q_\theta(r_t \mid s_t) \end{aligned} \tag{1}$$

Model Learning Dreamer (Hafner et al., 2019) makes future predictions in a feature space that is supervised by three signals: (a) image reconstruction $\left[\mathcal{J}_{\mathrm{O}}^t \doteq \ln q(o_t \mid s_t)\right]$, (b) reward prediction $\left[\mathcal{J}_{\mathrm{R}}^t \doteq \ln q(r_t \mid s_t)\right]$ and (c) dynamics regularization $\Big[\mathcal{J}_{\mathrm{D}}^t \doteq -\beta \mathrm{KL}\big(p(s_t \mid s_{t-1}, a_{t-1}, o_t) \big\| q(s_t \mid s_{t-1}, a_{t-1})\big)\Big]$. The overall objective is:

$$\mathcal{J}_{\text{Dreamer}} \doteq \mathbb{E}_\tau \Big[\sum_t \mathcal{J}_{\mathrm{O}}^t + \mathcal{J}_{\mathrm{R}}^t + \mathcal{J}_{\mathrm{D}}^t \Big] \tag{2}$$

optimized over the agent's experience τ. To achieve competitive performance on Atari, a few modifications are required that are incorporated in the variant *DreamerV2* described in (Hafner et al., 2020a).

Policy Learning Dreamer uses the learned forward dynamics model to train a policy using an actor-critic formulation described below:

$$\begin{aligned} &\text{Action model: } a_\tau \sim q_\phi(a_\tau \mid s_\tau) \\ &\text{Value model: } v_\psi(s_\tau) \approx \mathbb{E}q(\cdot|s_\tau) \textstyle\sum_{\tau=t}^{t+H} \gamma^{\tau-t} r_\tau \end{aligned} \tag{3}$$

The action model is trained to maximize cumulative rewards over a fixed horizon H. Both the action and value models are learned using imagined rollouts from the forward dynamics model. We refer the reader to (Hafner et al., 2020b) for more details.

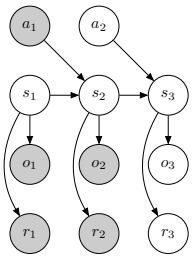
3 LEARNING TASK INFORMED ABSTRACTIONS

Task Informed MDP In the real world, the state space of an MDP is not directly observed. The graphical model corresponding to the MDP representing this scenario is shown in Figure 2a. As discussed earlier, visual observations typically contain both task-relevant and irrelevant information. In order to explicitly segregate these factors, we propose to factor the state space $\mathcal{S}$ into two components: a task-relevant component $\mathcal{S}^+$ and a task-irrelevant component $\mathcal{S}^-$. We assume that the reward is fully determined by the task-relevant component $r : \mathcal{S}^+ \mapsto \mathbb{R}$, and the task-irrelevant component contains no information about the reward: $\mathrm{MI}(r_t; s_t^-) = 0$ at each time step t.

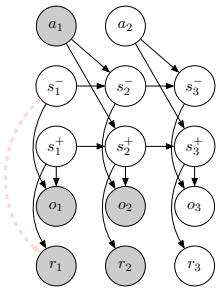
In the most general case, s_{t+1}^- can depend on s_t^+ and s_{t+1}^+ can depend on s_t^-. However, because we want separation between task relevant and irrelevant information, such dependence should be discouraged. We do so by making additional assumption of $p(s_{t+1}|s_t, a_t) = p(s_{t+1}^+|s_t^+, a_t)p(s_{t+1}^-|s_t^-, a_t)$. The resulting MDP, which we call **Task Informed MDP** (TiMDP) is illustrated in Figure 2b. Note that both subspaces $\mathcal{S}^+$ and $\mathcal{S}^-$ contribute to generate observation $\mathcal{O}$, and both forward models $p(s_{t+1}^+|s_t^+, a_t)$ and $p(s_{t+1}^-|s_t^-, a_t)$ admit the agent's actions. For clarity, we summarize the assumptions we have made into Table 2c.

Our method involves learning two models: one model captures the task-relevant state component s_t^+, which we call the *primary model*. The other model captures the task-irrelevant state component s_t^-, which we call the *secondary model*. The learning objective for these two models are denoted by $\mathcal{J}_{\mathrm{P}}$ and $\mathcal{J}_{\mathrm{S}}$ (primary and secondary), and expanded in Equation (4). A visual illustration is provided

(a) MDP



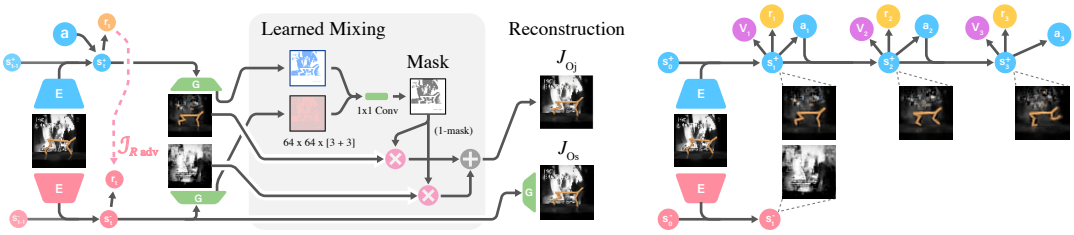
(b) Task-Informed MDP



(c) Assumptions for TiMDP

TiMDP	Details (dynamics)
$o_t = f(s_t^+, s_t^-)$	Both $\mathcal{S}^{-/+}$ contribute to $\mathcal{O}$
$r : \mathcal{S}^+ \mapsto \mathbb{R}$	r only depends on $\mathcal{S}^+$
$\mathrm{MI}(r_t; s_t^-) = 0$	$\mathcal{S}^-$ does not inform the task
$p(s_{t+1}\|s_t, a_t) = p(s_{t+1}^+\|s_t^+, a_t)\, p(s_{t+1}^-\|s_t^-, a_t)$	$s_{t+1}^{+/-}$ has no dependency on $s_t^{-/+}$

Figure 2: (a) The graphical model of an MDP. (b) Task-Informed MDP (TiMDP). The state space decomposes into two components: s_t^+ captures the task-relevant features, whereas s_t^- captures the task-irrelevant features. The cross-terms between $s^{+/-}$ are removed by imposing a factored MDP inductive bias. The red arrow indicates an adversarial loss to discourage s^- from picking up reward relevant information.



(a) Learning **Task Informed World Models** (b) **Policy Learning** only unrolls in $\mathcal{S}^+$.

Figure 3: Components of Task Informed Abstraction Learning. (a) From the dataset of past experience, TIA uses the reward to factor the MDP into a task-relevant world model and a task-irrelevant one. (b) Only the forward dynamics in s_t^+ is used during policy learning. The Policy is trained using back-propagation through time. Note that the images are shown just for demonstration purposes and are not generated during policy learning.

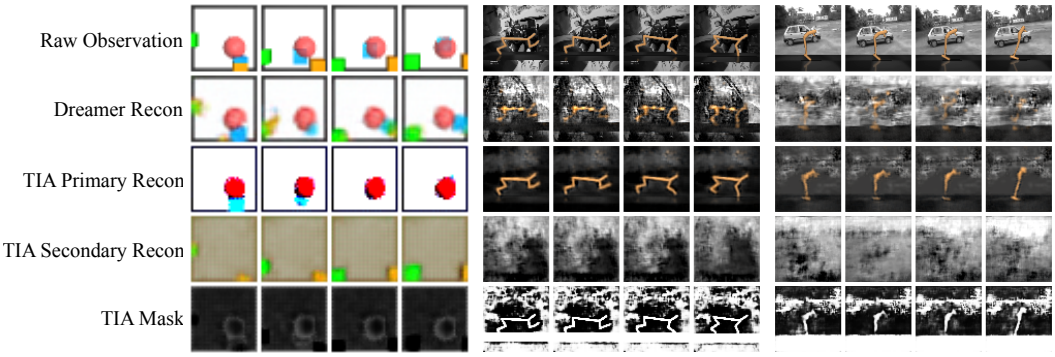
in Figure 3a.

$$
\begin{aligned}
&\mathcal{J}_{\mathrm{P}} \doteq \mathbb{E}_p\Big(\sum_t \Big(\mathcal{J}_{\mathrm{Oj}}^t + \mathcal{J}_{\mathrm{R}}^t + \mathcal{J}_{\mathrm{D}}^t\Big)\Big) \quad \mathcal{J}_{\mathrm{S}} \doteq \mathbb{E}_p\Big(\sum_t \Big(\mathcal{J}_{\mathrm{Oj}}^t + \mathcal{J}_{\mathrm{Os}}^t + \mathcal{J}_{\mathrm{Radv}}^t + \mathcal{J}_{\mathrm{Ds}}^t\Big)\Big) \\
&\mathcal{J}_{\mathrm{Oj}}^t \doteq \ln q(o_t \mid s_t^+, s_t^-) \qquad \mathcal{J}_{\mathrm{Os}}^t \doteq \lambda_{Os} \ln q(o_t \mid s_t^-) \\
&\mathcal{J}_{\mathrm{R}}^t \doteq \ln q(r_t \mid s_t^+) \qquad \mathcal{J}_{\mathrm{Radv}}^t \doteq -\lambda_{\mathrm{Radv}} \max_q \ln q(r_t \mid s_t^-) \\
&\mathcal{J}_{\mathrm{D}}^t \doteq -\beta D_{\mathrm{KL}} p(s_t^+ \mid s_{t-1}^+, a_{t-1}, o_t) q(s_t^+ \mid s_{t-1}^+, a_{t-1}) \\
&\mathcal{J}_{\mathrm{Ds}}^t \doteq -\beta D_{\mathrm{KL}} p(s_t^- \mid s_{t-1}^-, a_{t-1}, o_t) q(s_t^- \mid s_{t-1}^-, a_{t-1})
\end{aligned}
\tag{4}
$$

We will explain each component in the following section.

Reward Dissociation for the secondary model is accomplished via the adversarial objective $\mathcal{J}_{\mathrm{Radv}}^t$. This is a minimax setup where we interleave optimizing the secondary model's reward prediction head (for multiple iterations/training step) with training the secondary model. While the reward prediction head is trained to minimize the reward prediction loss $-\ln q(r_t|s_t^-)$, the secondary model maximizes this objective, so as to exclude reward-correlated information from its learned features. The reward prediction loss is computed using $\ln \mathcal{N}(r_t; \hat{r}_t, 1)$, where $\mathcal{N}(\cdot; \mu, 1)$ is the unit Gaussian, and $\hat{r}_t$ is the predicted reward.

Cooperative Reconstruction By jointly reconstructing the image, the secondary model that's biased towards capturing task-irrelevant information will enable the primary model to focus on



(a) ManyWorld (b) Cheetah Run (c) Hopper Stand

Figure 4: Visualizing the information represented by Dreamer Hafner et al. (2020b) and the primary and secondary models of our method on several environments. (a) In the ManyWorld environments, Dreamer mistakes the distractor (yellow) for the target object (blue). The primary model of TIA isolates the target object (blue) and the goal (red). (b, c) Dreamer's capacity is consumed at reconstructing the irrelevant video background, and it fails to capture the agent's outline, which is the task-relevant information. In all domains, Dreamer reconstruction tries to capture every pixel of the raw observation but misses task-relevant information. TIA is able to capture task-relevant information with the primary model and task-irrelevant information with the secondary model.

task-relevant features. We implement joint reconstruction through the objective $\mathcal{J}^t_{\text{Oj}}$. Starting with a sequence of observation and actions $\{o_{[<t]}, a_{[<t]}\}$, we first pass this sequence through the two separate recurrent state space model (RSSM) to produce the activation $h^+_t$and h^-_t, which are concatenated with the observation embedding from the two encoders separately to decode two images $\hat{o}^+_t$ and $\hat{o}^-_t$. Given the observation o_t, the joint reconstruction is achieved through a learned mixing where each model additionally decodes a $64 \times 64 \times 3$ tensor. These two tensors are concatenated channel-wise before being passed through a 1×1 convolution layer followed by sigmoid activation to obtain a $64 \times 64 \times 1$ mask M_t with the value between $(0, 1)$. The final reconstruction is obtained through $\hat{o}_t = \hat{o}^+_t \odot M_t + \hat{o}^-_t \odot (1 - M_t)$, where $\odot$ denotes element-wise product. The reconstruction objective is computed as $\ln \mathcal{N}(o_t; \hat{o}_t, 1)$.

Secondary-only Reconstruction A degenerate case exists where a secondary model that captures no information at all can still satisfy the two objectives above, letting the primary model reconstruct the entire observation by itself. To avoid such degeneracy, we add an additional image decoder to encourage the secondary model to capture as much information from the observation as possible. This is implemented via the objective $\mathcal{J}^t_{Os}$.

Policy Learning is similar to Dreamer, except that we replace the world model with the primary model. This way, the forward unroll only involves the $\mathcal{S}^+$ subspace. The policy takes the latent vector h_t as an input:

$$\begin{aligned} &\text{Action model: } a_\tau \sim \; q_\phi(a_\tau \mid s^+_\tau) \\ &\text{Value model: } \; v_\psi(s^+_\tau) \approx \mathbb{E}_{q(\cdot|s^+_\tau)}(\textstyle\sum_{\tau=t}^{t+H} \gamma^{\tau-t} r_\tau) \end{aligned} \tag{5}$$

An illustration of the policy learning stage is in Figure 3b.

4 Experiments

Our empirical evaluation aims to answer if our method outperforms existing methods when learning in environments that contain irrelevant information in the form of distractor objects or visually complex backgrounds. For this purpose we make use of three environments that are described in Section 4.1. Baseline wise, we compare our method against several alternatives described in Section 4.2.

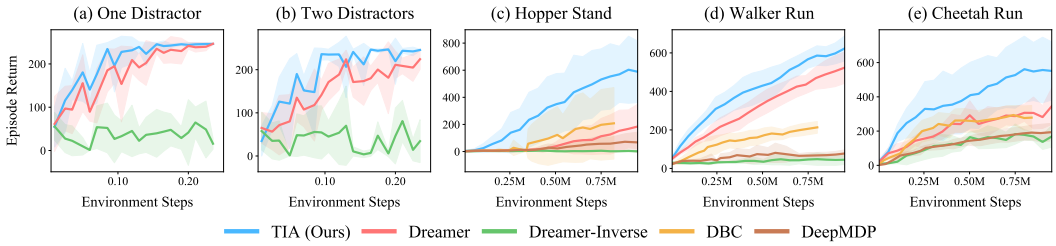


Figure 5: Our method consistently outperforms Dreamer and other baseline methods in a variety of visual control tasks with distraction. The curves show mean and standard deviation, over five seeds for TIA, Dreamer, and Dreamer-Inverse. Results for DBC and DeepMDP are adopted from results reported in Zhang et al. (2020b) and used ten seeds. Our method is effective for both ManyWorld environments (a,b), which contains confusing distracting objects that look similar to the task-relevant components; and the DMC tasks with natural video backgrounds (c,d,e), where the distracting background contains rich information that would consume significant model capacity to capture.

4.1 Environments

ManyWorld (Figure 4a) We want a test environment where one can vary the level of distraction in a controlled manner. For this purpose, we introduce ManyWorld, a physics domain where one can control the number of objects and their dynamics. The task is to move a target block (in blue) to a location indicated visually by a translucent red sphere. Other objects act as distractors. We turn off collision between objects so that they do not physically interfere with the target object, but occlusion does occur. The visual similarity between the objects introduces confusion, requiring additional effort to resolve when learning a world model.

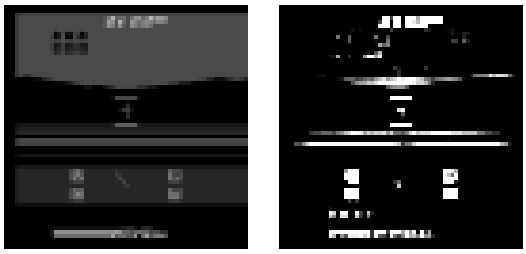


Figure 6: (Left) Raw observation of ATARI Robotank. (Right) The primary model of TIA emphasizes task-relevant information such as the crosshair and the radar for tracking enemies, while ignoring task-irrelevant information such as textures in the raw observation.

Kinematic Control with Natural Video Distraction (Figure 4b) We consider the DeepMind Control (DMC) suite with natural video background from the Kinetics dataset Kay et al. (2017) used in prior work Zhang et al. (2020b), which was introduced specifically to test learning under natural visual distraction. These control tasks involve different types of challenges, such as long planning horizon (Hopper), contact and collision (Walker), and larger state/action space (Cheetah). The natural video backgrounds in this test suite contain a large number of factors of variation, adding additional burden to world-model learning and negatively affecting policy optimization downstream. It is a challenging domain for both model-based and model-free algorithms to master. The video backgrounds are from the class "driving car" and are grayscale as in Zhang et al. (2020b).

Atari Learning Environments (ALE) (Figure 6) are a standard benchmark for vision-based control. The visuals of these games naturally contain many distractor objects that are irrelevant to the game objective. Our limited compute resources only allowed us to experiment on five games. Each seed takes a week on a V100 Volta GPU. We present results on games where state-of-the-art model-based algorithms perform significantly worse than model-free algorithms or human performance in the hope of closing this gap.

4.2 Baseline Methods

We include both model-based and model-free baselines, a few proposed specifically to tackle learning in the presence of task-irrelevant distractions. In particular we compare against **Dreamer** Hafner et al. (2020b) which is a state of the art model-based algorithm on DMC. On ALE, we compare against an improved variant **Dreamer(V2)** Hafner et al. (2020a). We compare against a strong model-free

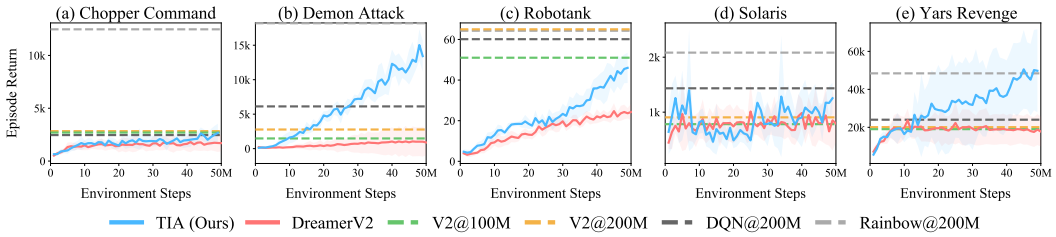


Figure 7: ATARI performance at 50M steps. TIA significantly improves policy learning in Demon Attack, Robotank, and Yars Revenge, in which DreamerV2 fails to learn or has inferior sample complexity. We also add the performance of DreamerV2 at 100M and 200M steps, along with the performance of two model-free algorithms, DQN and Rainbow DQN, at 200M steps.

method, **Deep Bisimulation for Control (DBC)** Zhang et al. (2020b), which uses the bisimulation metric and is developed specifically to tackle task-irrelevant distractions. Finally, we also include **DeepMDP** Gelada et al. (2019) which learns a forward model with the sole purpose of acquiring a representation, then uses model-free, distributional Q learning for the policy. The DeepMDP and DBC results are adapted from Zhang et al. (2020b).

Representation learning through an inverse model Inverse model take the observations o_t and o_{t+1} as input and predict the intervening action a_t Agrawal et al. (2015); Jayaraman & Grauman (2015). To investigate if features learned by inverse model suffice for learning task-relevant features, we constructed the *Dreamer-Inverse* model. In this model, the learning objective becomes the following: $\mathcal{J}^t_{\text{Inverse}} = \mathcal{J}^t_{\text{inv}} + \mathcal{J}^t_R + \beta \mathcal{J}^t_D$ where $\mathcal{J}^t_{\text{inv}} \doteq \ln q(a_t \mid s_t, s_{t+1})$ is the inverse model and $\mathcal{J}^t_{\text{D}}$ is the dynamics regularizer described in Section 2.

4.3 Can TIA disassociate task-irrelevant information?

We first evaluate our method on the ManyWorld domain, where relevant information comprises the agent (blue block) and the goal (red sphere). Figure 4a provides a qualitative comparison of the information represented by TIA and the baseline method of *Dreamer*. In many cases, *Dreamer* mistakes the distractor (yellow) for the agent (blue). On the other hand, the primary model of TIA isolates the agent (blue) and the goal (red), while the secondary model of TIA captures the distractors (yellow and green). This demonstrates that indeed s_t^+ captures task-relevant information and successfully ignores the rest. Quantitative evaluation depicted in Figure 5 (a) and (b) demonstrates that TIA outperforms the baselines, and the performance gap increases with the number of distractors.

Next, we considered the DMC domains with natural video distractions. We evaluate on three environments, *Cheetah Run*, *Walker Run* and *Hopper Stand* since they are visually different and cover a variety of learning challenges described in Section 4.1. The image reconstruction results Figure 4b and Figure 4c show that Dreamer performs poorly in capturing the full state of the agents and is distracted by the background. In contrast, the primary model of our TIA method accurately recovers the relevant part of the raw observation, which happens to be the agent's body in these examples. Quantitative performance reported in Figure 5 (c,d,e) clearly shows that TIA outperforms strong baseline methods described in Section 4.2. Overall, these results suggest that TIA is the new state-of-the-art in learning from cluttered observations.

4.4 Results on ATARI

In previous test environments, the irrelevant factors were manually injected. Due to human intervention, it is possible these tasks were biased, which our method exploited. To investigate performance in more natural settings, we experimented on ATARI games that innately contain significant visual distractions. E.g., images of *robotank* game contain several visual signatures that change during the game: the number of enemy tanks destroyed, the rotating radar scanner, the green sprites, and so on. We evaluated performance on five ATARI games that are known to be challenging for model-based methods Hafner et al. (2020b) without any hyperparameter tuning (i.e., just a single value chosen based on the intuition described in Section 4.5).

The results reported in Figure 7 demonstrate that we substantially outperform the strong baseline of DreamerV2. Furthermore, for the games of *Chopper Command*, *Demon Attack* and *Yars Revenge* we either match or outperform strong model-free baselines of DQN/Rainbow trained for 200M steps, while our method is only trained for 50M steps. These results convincingly demonstrate the superiority of our method. Figure 6 shows the image reconstruction of TIA's primary model in *Robotank*.

4.5 Hyperparameter Selection

The two important hyperparameters are λ_{Radv} and λ_{Os}. One particular mode of failure is when the secondary model takes over the reconstruction. It strips the primary model from task-relevant information, thus preventing the policy from learning any meaningful behavior. Our reward dissociation scheme relies on informative reward signals to work. Yet, at the beginning of training, the reward collected by a random policy tend to be sparse and noisy, making λ_{Radv} less effective at preventing a dominant secondary model. This scenario suggests using a large λ_{Radv} at the beginning of training and slowly increasing the weight λ_{Os} for the secondary reconstruction loss.

The other extreme is for the secondary model to collapse into degeneracy, where it fails to capture any information from the observations. In this case, TIA degenerates into Dreamer, and we can increase λ_{Os} so that the secondary model is encouraged to capture more.

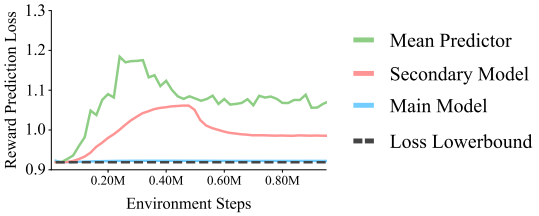


Figure 8: **Reward Dissociation During Learning** We plot the negative log-likelihood loss of the reward prediction $-\ln p(r)$, (lower bounded by 0.92) of a mean predictor, the reward prediction module of the primary model, and the reward prediction module of the secondary model. The features from the primary model contain sufficient information for reward prediction. The performance of the reward predictor for the secondary model follows the same trend as the mean predictor, indicating that the features learned by the secondary model are reward-independent.

To gain more insight into the reward dissociation process, we want to know how much information of the reward does the secondary model capture during learning. We use errors in predicting the reward, measured with the log-likelihood (as unit Gaussian) in Figure 8. We estimate the upper bound of the prediction error using the trailing average of the reward. This corresponds to an uninformed reward predictor (the "Mean Predictor") that always guesses the average. The reward prediction loss of the secondary model trails behind this upper bound during learning (see Figure 8, in red), whereas the reward prediction loss of the main model remains small (in blue), slightly above the loss lower bound, which equals to $-\ln \mathcal{N}(0;\ 0, 1) = -\ln \frac{1}{\sqrt{2\pi}} \approx 0.92$.

5 Conclusion and Discussion

In this work, we have shown that the TiMDP formulation that explains away task-irrelevant information can successfully learn from cluttered visual inputs. Our approach of learning **T**ask **I**nformed **A**bstractions (TIA), outperforms previous state-of-the-art model-based RL methods on multiple standard benchmarks.

An issue worth mentioning is that while one set of hyperparameters worked well across ATARI games, for the DMC environments, the choice of hyperparameters, $\lambda_{\text{Radv}}, \lambda_{\text{Os}}$, was domain-dependent. We discussed good practices for choosing these hyperparameters in Section 4.5. We hypothesize that based on these practices, in the future, it might be possible to automatically tune the hyperparameters by considering the reconstruction and reward-prediction loss of the two models that constitute TIA.

Our goal in this work was to build agents that operate from complex visual imagery. While we outperform previous methods, all of our evaluation is on simulated data. We plan to test our method on real-world data in the future. Another area of potential investigation is to characterize performance as a function of the sparsity of reward signals. We hypothesize that a low-noise reward signal is a key factor for the robustness of TIA, and the performance might drop in very sparse reward scenarios. Developing methods that can overcome this "potential" challenge is another avenue for future work.

REFERENCES

Rishabh Agarwal, Marlos C Machado, Pablo Samuel Castro, and Marc G Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. *arXiv preprint arXiv:2101.05265*, 2021.

Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *Proceedings of the IEEE international conference on computer vision*, pp. 37–45, 2015.

Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. In *Advances in Neural Information Processing Systems*, pp. 5074–5082, 2016.

Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *International Conference on Machine Learning*, pp. 528–539. PMLR, 2020.

Ershad Banijamali, Rui Shu, Hung Bui, Ali Ghodsi, et al. Robust locally-linear controllable embedding. In *International Conference on Artificial Intelligence and Statistics*, pp. 1751–1759. PMLR, 2018.

Yoshua Bengio. Deep learning of representations: Looking forward. May 2013. URL http://arxiv.org/abs/1305.0445.

Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. In *Advances in neural information processing systems*, pp. 841–852, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 4067–4080. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1418. URL https://doi.org/10.18653/v1/D19-1418.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Learning to model and ignore dataset bias with mixed capacity ensembles. *arXiv preprint arXiv:2011.03856*, 2020.

Thomas L. Dean, Robert Givan, and Sonia M. Leach. Model reduction techniques for computing approximately optimal solutions for markov decision processes. In Dan Geiger and Prakash P. Shenoy (eds.), *UAI '97: Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, Brown University, Providence, Rhode Island, USA, August 1-3, 1997*, pp. 124–131. Morgan Kaufmann, 1997. URL https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=303&proceeding_id=13.

Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. Provably efficient rl with rich observations via latent state decoding. In *International Conference on Machine Learning*, pp. 1665–1674. PMLR, 2019.

Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I Jordan, Joseph E Gonzalez, and Sergey Levine. Model-based value estimation for efficient model-free reinforcement learning. *arXiv preprint arXiv:1803.00101*, 2018.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *UAI*, volume 4, pp. 162–169, 2004.

Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation metrics for continuous markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011.

Norman Ferns and Doina Precup. Bisimulation metrics are optimal value functions. In *UAI*, pp. 210–219. Citeseer, 2014.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2786–2793. IEEE, 2017.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. Deepmdp: Learning continuous latent space models for representation learning. *arXiv preprint arXiv:1906.02736*, 2019.

Robert Givan, Thomas L. Dean, and Matthew Greig. Equivalence notions and model minimization in markov decision processes. *Artif. Intell.*, 147(1-2):163–223, 2003. doi: 10.1016/S0004-3702(02)00376-4. URL https://doi.org/10.1016/S0004-3702(02)00376-4.

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020a.

Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020b. URL https://openreview.net/forum?id=S1lOTC4tDS.

He He, Sheng Zha, and Haohan Wang. Unlearn dataset bias in natural language inference by fitting the residual. *arXiv preprint arXiv:1908.10763*, 2019.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. Darla: Improving zero-shot transfer in reinforcement learning. In *International Conference on Machine Learning*, pp. 1480–1490. PMLR, 2017.

Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to ego-motion. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1413–1421, 2015.

Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H. Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model based reinforcement learning for atari. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=S1xCPJHtDB.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

Diederik P Kingma and Max Welling. Stochastic gradient vb and the variational auto-encoder. In *Second International Conference on Learning Representations, ICLR*, volume 19, 2014. URL https://pdfs.semanticscholar.org/60ea/adac49b19d1f41925df3bbbb7983a113acec.pdf.

Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In Z Ghahramani, M Welling, C Cortes, N D Lawrence, and K Q Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, pp. 3581–3589. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5352-semi-supervised-learning-with-deep-generative-models.pdf.

Alexander S Klyubin, Daniel Polani, and Chrystopher L Nehaniv. Empowerment: A universal agent-centric measure of control. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pp. 128–135. IEEE, 2005.

Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e615c82aba461681ade82da2da38004a-Abstract.html.

Kuang-Huei Lee, Ian Fischer, Anthony Liu, Yijie Guo, Honglak Lee, John Canny, and Sergio Guadarrama. Predictive information accelerates learning in rl. *arXiv preprint arXiv:2007.12401*, 2020.

Bogdan Mazoure, Remi Tachet des Combes, Thang Doan, Philip Bachman, and R Devon Hjelm. Deep reinforcement and infomax learning. *arXiv preprint arXiv:2006.07217*, 2020.

R McCallum. Reinforcement learning with selective perception and hidden state. 1997. URL https://urresearch.rochester.edu/institutionalPublicationPublicView.action?institutionalItemId=678.

Junhyuk Oh, Satinder Singh, and Honglak Lee. Value prediction network. July 2017. URL http://arxiv.org/abs/1707.03497.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 2778–2787, 2017.

Sébastien Racanière, Théophane Weber, David P Reichert, Lars Buesing, Arthur Guez, Danilo Rezende, Adria Puigdomenech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, et al. Imagination-augmented agents for deep reinforcement learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 5694–5705, 2017.

Roberta Raileanu, Max Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. Automatic data augmentation for generalization in deep reinforcement learning. *arXiv preprint arXiv:2006.12862*, 2020.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *arXiv preprint arXiv:1911.08265*, 2019.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1134–1141. IEEE, 2018.

David Silver, Hado van Hasselt, Matteo Hessel, Tom Schaul, Arthur Guez, Tim Harley, Gabriel Dulac-Arnold, David Reichert, Neil Rabinowitz, Andre Barreto, and Thomas Degris. The predictron: End-to-end learning and planning. December 2016. URL http://arxiv.org/abs/1612.08810.

Aravind Srinivas, Allan Jabri, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Universal planning networks: Learning generalizable representations for visuomotor control. In *International Conference on Machine Learning*, pp. 4732–4741. PMLR, 2018.

Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

Austin Stone, Oscar Ramirez, Kurt Konolige, and Rico Jonschkowski. The distracting control suite–a challenging benchmark for reinforcement learning from pixels. *arXiv preprint arXiv:2101.02722*, 2021.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. *arXiv preprint arXiv:2009.08319*, 2020.

Aviv Tamar, Yi Wu, Garrett Thomas, Sergey Levine, and Pieter Abbeel. Value iteration networks. *arXiv preprint arXiv:1602.02867*, 2016.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. Deepmind control suite. January 2018. URL http://arxiv.org/abs/1801.00690.

Niklas Wahlström, Thomas B Schön, and Marc Peter Deisenroth. From pixels to torques: Policy learning with deep dynamical models. *arXiv preprint arXiv:1502.02251*, 2015.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pp. 10506–10518, 2019.

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *Advances in neural information processing systems*, 28:2746–2754, 2015.

Wilson Yan, Ashwin Vangipuram, Pieter Abbeel, and Lerrel Pinto. Learning predictive representations for deformable objects using contrastive estimation. *arXiv preprint arXiv:2003.05436*, 2020.

Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. October 2019. URL http://arxiv.org/abs/1910.01741.

Amy Zhang, Clare Lyle, Shagun Sodhani, Angelos Filos, Marta Kwiatkowska, Joelle Pineau, Yarin Gal, and Doina Precup. Invariant causal prediction for block mdps. In *International Conference on Machine Learning*, pp. 11214–11224. PMLR, 2020a.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020b.

APPENDIX

A OTHER RELATED WORKS

State-of-the-art deep reinforcement learning algorithms often jointly optimize the the discounted return together with an auxiliary representation learning objective such as image reconstruction Watter et al. (2015); Wahlström et al. (2015) or contrastive predictive coding (CPC, Sermanet et al. 2018; Oord et al. 2018; Yan et al. 2020; Lee et al. 2020; Mazoure et al. 2020). In model-based reinforcement learning methods, reward and value prediction Oh et al. (2017); Racanière et al. (2017); Silver et al. (2016); Tamar et al. (2016); Feinberg et al. (2018); Schrittwieser et al. (2019) are also shown to improve performance, along with learning a world model. More recently, data augmentations are found to improve sample complexity Laskin et al. (2020); Kostrikov et al. (2020); Raileanu et al. (2020); Srinivas et al. (2020); Stooke et al. (2020) by taking advantage of domain knowledge of symmetry transformations in the space of data. A recent benchmark Stone et al. (2021) however,

shows that data augmentation helps but still performs poorly in the presence of complex visual distractions. We do find data augmentation to be an orthogonal and complementary approach to our proposal, which focuses on the way to inform representation learning for RL, which feature is more useful, and therefore should be learned first.

A more principled approach exploits additional structure in the real world to learn state-abstractions. The most representative are block MDPs (Du et al., 2019), bisimulation Givan et al. (2003); Dean et al. (1997) and bisimulation metrics Ferns et al. (2004; 2011); Ferns & Precup (2014), including recent impressive empirical gains on natural scene domainsZhang et al. (2020b); Gelada et al. (2019); Agarwal et al. (2021). The work presented here fall under the broad umbrella of learning state and temporal abstractions and is the most closely related to *utile distinction* McCallum (1997) which is limited to finite-state machines. Like *utile distinction*, TIA is distinct from DBC and bisimulation embeddings in that we rely on the task specification for feature separation, and we are not concerned about generalization across a set of MDPs related by dynamics.

The spiritual nearest neighbor to our work is the idea of using a primary-bias duo for model debiasing in supervised learning Clark et al. (2019); Cadene et al. (2019); Wang et al. (2019); He et al. (2019); Bahng et al. (2020); Clark et al. (2020). These methods remove dataset bias by imposing independence constraints between the primary model and a model that's biased by design. Our approach focuses on RL and uses two models for removing task-irrelevant features for policy learning. Our implementation extends recent work in model-based RL from pixels Watter et al. (2015); Finn & Levine (2017); Banijamali et al. (2018); Ha & Schmidhuber (2018); Kaiser et al. (2020); Hafner et al. (2020b;a); Srinivas et al. (2018).

B Performance Gap When Learning In The Presence of Visual Distractions

When complex visual distractors are present in the observations, state of the art model-based agents struggle to maintain their original sample efficiency and asymptotic performance. We produce this gap in details with Figure 9, where we include three additional (six in total) domains from the DeepMind Control suite. We arrange the domains in the order of difficulty levels, from easy to hard for the model-based approach. This gap is smaller on the two easiest domains, *Walker stand* and *Walker walk*; and is much more pronounced in *Walker run*, *Hopper stand*, and *Cheetah run*. On the most challenging task, *Finger spin*, where the model-based algorithm dreamer Hafner et al. (2020b) performs much worse than model-free approaches, the gap almost disappears – making the domain ill-suited for testing our proposal. Note that the maximum episodic return on these tasks is calibrated to 1000 Tassa et al. (2018). We additionally provide the ratio for the number of pixels that are replaced by the background video for each domain, as a proxy for how much visual distraction is introduced: Walker 63%, Hopper 77%, Cheetah: 83%, and Finger 92%.

Our intent with this paper and the proposal, learning *Task Informed Abstractions*, is to close this gap, such that model-based agents can retain its performance even when learning in the presence of complex visual distractions.

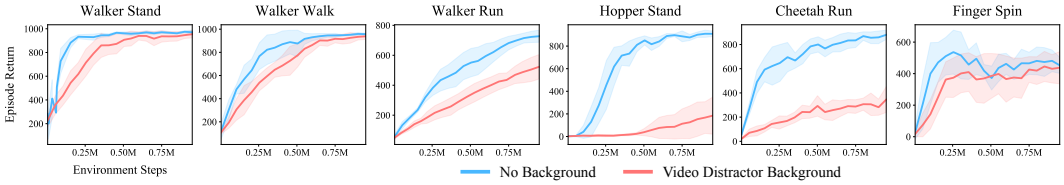


Figure 9: **The data efficiency and performance gap** when learning with video distraction backgrounds.

B.1 Bridging the Gap by Learning Task Informed Abstractions

We produce the complete result on DeepMind Control Suite including the 3 additional tasks from above where we only expect marginal improvements. We setup the experiments by matching the total number of parameters, so that each one of TIA's two models are only half in size as the single world

model from dreamer. This comparison is disadvantage to our method, because the smaller primary model runs the risk of being too small to capture the set of task-relevant features in its entirety.

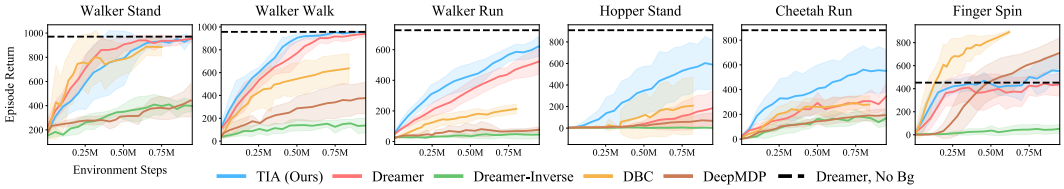


Figure 10: **Performance of Task Informed Abstractions** on three additional DeepMind control domains.

Despite of this disadvantage in model capacity, task informed abstraction is able to reduce the gap on *Walker run* while making the gap significantly smaller on *Hopper stand* and *Cheetah run* (see Figure 9). On *Walker stand* and *Walker walk* the gap is small to begin with, therefore our method cannot bring much benefit. Finally, despite being a poor choice for testing our method, we include results on *Finger spin* for the sake of completeness.

C Understanding Failure Cases

In Section 4.5 we provide the principled approach to tuning the hyperapameters λ_{Radv} and λ_{Os}, where we balance these two terms to avoid either one of the two models taking over the entire reconstruction. We label these two extreme cases, where either the secondary model, or the primary model takes over, as type I, and type II. We provide detailed renderings of these failure cases below Figure 11, in comparison to a successfully learned world model.

The first column in Figure 11a shows a successfully trained agent whose primary model is able to perfectly reconstruct the walker agent. In type I failure mode (Figure 11b) the secondary model would take over the entire reconstruction, causing the primary model to lose its grasp on the task-relevant features. This would prevent the policy from learning any useful behavior, which makes the collected reward practically random. With random rewards, the term $\mathcal{J}_{\text{Radv}}$ is ineffective at dissociating task-relevant features, causing training to fail. The policy performance under this scenario is usually close to that of a random policy. When this happens, we want to increase λ_{Radv} to dissociate the secondary model from the reward, or decrease λ_{Os} that weighs the secondary reconstruction term.

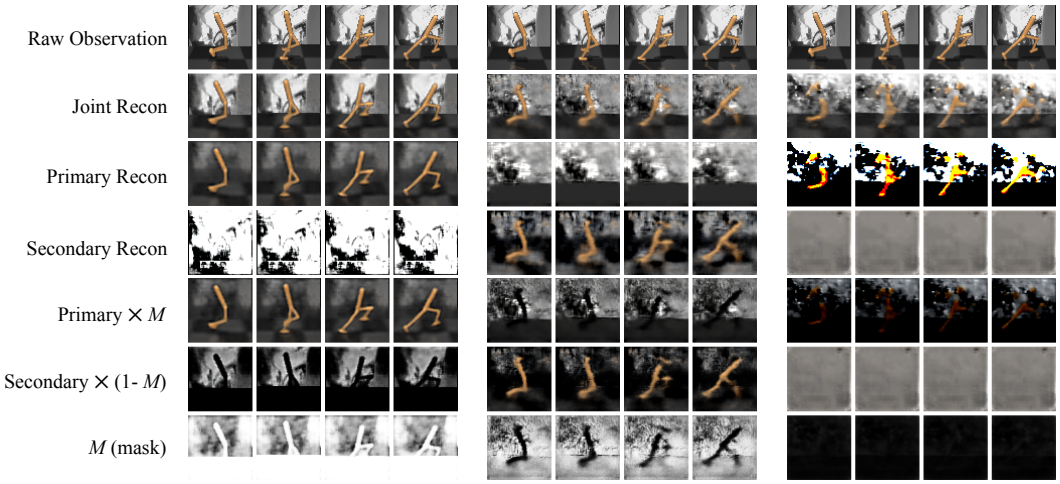
In the type II failure case (see Figure 11c), the primary model takes over the reconstruction. In this case the model degenerates into dreamer without separating out the task-irrelevant features, and the performance is close to a dreamer agent with a smaller model. In this case we would increase the weight λ_{Os} on the secondary reconstruction to encourage it to learn more features.

By tuning these two parameters λ_{Radv} and λ_{Os}, we were able to avoid these two failure modes, and stabilize training on a wide variety of domains. A future direction would be to tune these to parameters automatically using the signals mentioned in Section 4.5.

D Model Sizes And Architecture Details

For fair comparison, on DMC and ManyWorld we match the total number of parameters of our two world models combined with that of a single, large Dreamer model. The total number of parameters is 10 million on DMC, and 1 million on ManyWorld. We divide the two models equally in size, with an additional image reconstruction head for the secondary model. *Dreamer-Inverse* has the same size for all model components as the Dreamer model except it replaces the deconvolution head with an action prediction head for learning the inverse dynamics. We scale the model size by changing the width of each layer in the networks, without changing the overall architecture or the depth of the networks. All other hyperparameters such as learning rates are kept the same as Hafner et al. (2020b).

On the ATARI Learning Environments we compare against the state-of-the-art on this domain, DreamerV2 Hafner et al. (2020a), which uses a world model of 20 million trainable parameters. We made the primary model the same size as the original implementation while adding a smaller



(a) Success

(b) Failure Type 1: The secondary model takes over in reconstruction

(c) Failure Type 2: The primary model takes over in reconstruction

Figure 11: Detailed rendering from one successfully trained walker agent, and two failure cases of type 1 and type 2. In type 1 failure case, the secondary model takes over the entire reconstruction, rendering the primary model ineffective for policy learning (random policy). In the type 2 failure case, the primary model attempts to capture all factors of variation by itself, thus failing to perfectly reconstruct the image, also leading to sub-par policy performance in the lower 400.

secondary model which contains 12 million parameters. A key difference between Dreamer and DreamerV2 is that the latter has an additional prediction head for the discount factor γ_t besides the standard reward prediction head. This discount factor head plays an instrumental role in allowing DreamerV2 to solve ATARI games, therefore we additionally dissociate the secondary model from information about the discount factor, by adding an additional adversarial prediction loss. We use the same scale for the discount factor γ as that for reward: $\lambda_{\gamma\text{adv}} = \lambda_{\text{Radv}}$. All other hyperparameters such as learning rates are kept the same as Hafner et al. (2020a).

We use an input size of $32\times32\times3$ in ManyWorld, $64\times64\times3$ in DeepMind Control Suite, and $64\times64\times1$ in ATARI games. We use grayscale for the natural video backgrounds, the same as previous work Zhang et al. (2020b).

E Hyperparameters

For fair comparison, we did not tweak existing hyperparameters from Dreamer and used identical settings as Hafner et al. (2020b) and Hafner et al. (2020a). Our reward-dissociation scheme introduces two new hyperparameters λ_{Radv} and λ_{Os}. We scale the reward dissociation loss via λ_{Radv} such that the term matches reconstruction losses in magnitude. For this reason, the differences in scale in Table 1 mostly reflect the differences in input image sizes. We tweaked λ_{Os} to stabilize training. Detailed settings for each domain are in Table 1.

F Transfer to Novel Distractions

The task informed abstraction we introduce in this paper improves learning when distractions are present. To adapt to out-of-distribution scenarios unseen during training, additional architectural changes that reject distracting image features on the fly may be required. To provide a baseline and intuitions for this future direction, we evaluate how well existing agents perform under this type of domain shift. In Table 2 we take agents that are trained (1) without video background, (2) with background videos from the *driving car* class or (3) with white noise backgrounds, and evaluate against background videos from a different class, *walking the dog* (labeled as *transfer*, see Table 2).

Table 1: Hyperparameters

Domain and Task	λ_{Radv}	λ_{Os}
ManyWorld, 1 Distractor	600.0	2.0
ManyWorld, 2 Distractor	150.0	2.0
Hopper Stand	30k	2.0
Cheetah Run	20k	1.5
Walker Run	20k	0.25
Walker Walk	20k	0.25
Walker Stand	20k	0.25
Finger Spin	30k	2.5
All ATARI games	2k	1.0

Table 2: **DeepMind Control Transfer Performance** transfer to the video class *walking the dog* as background

Training Condition		Drmr, No Bg	Drmr, Video	TIA, Video	Drmr, Noise	TIA, Noise
Hopper Stand	In-domain	906.8 ± 29.3	183.8 ± 162.1	596.4 ± 234.1	769.7 ± 205.4	744.8 ± 75.8
	Transfer	18.1 ± 12.1	186.2 ± 142.2	629.4 ± 231.5	357.4 ± 206.7	354.9 ± 136.9
Walker Run	In-domain	728.2 ± 37.8	520.2 ± 84.4	625.3 ± 64.7	737.6 ± 26.7	696.9 ± 43.9
	Transfer	127.4 ± 34.0	530.9 ± 76.9	645.3 ± 78.4	341.1 ± 108.9	345.5 ± 138.0
Cheetah Run	In-domain	876.3 ± 36.0	325.7 ± 96.6	556.6 ± 167.7	754.9 ± 67.0	734.2 ± 163.4
	Transfer	21.1 ± 7.7	312.6 ± 115.7	557.4 ± 194.6	227.0 ± 75.1	309.5 ± 233.2

The Dreamer agent trained with no background distraction fails to transfer its performance when background videos are introduced at test time, which is expected. In the second experiment we train both dreamer and TIA with video background, but test using videos from a different category. Dreamer did not learn as well as TIA as indicated by its poor performance in the training environments, but both methods retain their training performance post-transfer, unaffected by the change in the background video. As control, we also train both methods using white noise as the background. The training and transfer performance are both identical between the two methods, and the transfer performance is worse than performance on white noise.

We additionally evaluate ManyWorld agents that are trained with (1) no distractor, (2) one distractor, or (3) two distractors, with an additional distractor (three distractors, see Table 3).

Table 3: **ManyWorld Transfer Performance** transfer to three distraction blocks.

Training Condition		Drmr, 0	Drmr, 1	Drmr, 2	TIA, 1	TIA, 2
ManyWorld	In-domain	246.0 ± 3.5	242.9 ± 5.4	217.4 ± 29.3	246.1 ± 1.7	245.8 ± 1.8
	Transfer	192.6 ± 18.9	198.4 ± 27.4	192.0 ± 35.1	185.7 ± 21.4	233.4 ± 6.5

Both results show that while TIA learns better from cluttered scenes, mechanism to reject unseen backgrounds at decision time is required to transfer successfully. This points to the incorporation of attention as a great avenue for future work.