Learning Task Informed Abstractions

Learning Task Informed Abstractions

Xiang Fu [*1] Ge Yang [*1 2] Pulkit Agrawal [1 2] Tommi Jaakkola [1]

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 explicitly separates reward-correlated visual features from distractors. For learning TIA, we introduce the formalism of Task Informed MDP (TiMDP) that 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. Project page: https://xiangfu.co/tia

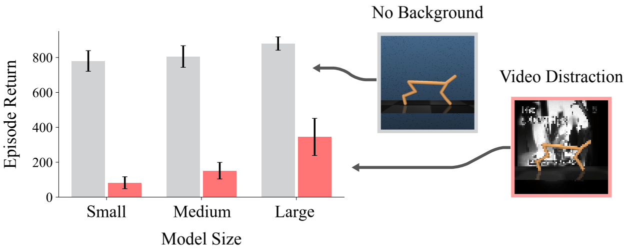


Figure 1: Comparison of the performance of a state-of-the-art model-based RL algorithm, *Dreamer*, on two versions of the *Cheetah Run* with vs. without visual distraction. Performance is reported for three models of increasing sizes (0.5×, 1×, 2× of the original Dreamer). Results show that even the 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.

1. Introduction

Consider results of a simple experiment reported in Figure 1. We train a state-of-the-art model-based reinforcement learning algorithm (Hafner et al., 2020) 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., 2021). For each version we train three model variants that contain $0.5\times$ (small), $1\times$ (medium) and $2\times$ (large) parameters in their respective world models. The performance with the simple background is only marginally affected by model capacity indicating that even the smallest model has sufficient capacity to learn task-relevant features. With complex background, performance is much worse but increases monotonically with the model size. Because the amount of task-relevant information is unchanged between simple and complex background variants, these results demonstrate that excess model capacity is devoted to representing background information when learning from complex visual inputs. The background conveys no information about the task. It, therefore, interferes with the learning of task-relevant information by consuming model capacity. Here "relevant" refers to features needed to predict the optimal actions, whereas "irrelevant" refers to everything else that makes up the observation.

There are two main components of a model-based learner: (i) a forward dynamics model that predicts future events resulting from executing a sequence of actions from the current state and (ii) a reward predictor for evaluating possible future states. The policy performance depends critically on the prediction accuracy of the forward model, which is intimately tied to the feature space in which the future is predicted. Similar to the complex background version of *Cheetah Run*, visual observations obtained in the real world are full of irrelevant information. Therefore, if there is no bias for learning "task-relevant features", the model will try to predict all the information. In such a scenario spurious features would unnecessarily increase the sample complexity and necessitate the use of much larger models than necessary. Sometimes it can also result in training failure.

Ways to learn good representations has been a major focus in model-based deep reinforcement learning. A popular choice is to reconstruct the input observations (Kingma & Welling,

[*] Equal contribution [1] MIT CSAIL [2] IAIFI. Correspondence to: Xiang Fu <xiangfu@csail.mit.edu>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

2014; Kingma et al., 2014; Watter et al., 2015; Hafner et al., 2020). 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 space, this disentangled feature space would still contain irrelevant information and will not address the central problem of learning task-relevant features. As our analysis shows, the incorporation of reward prediction loss is insufficient for producing a feature space that only contains task-relevant information. Reward supervision alone is shown insufficient 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 the knowledge of the full-body pose might be necessary to predict 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., 2020; Oh et al., 2017) but 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., 2017; Oh et al., 2017; Schrittwieser et al., 2020). However, because the value function is often learned via bootstrapping, it may provide an unstable training signal.

These challenges in learning task-relevant representations 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., 2021). Because supervision in bisimulation comes solely from rewards, it is subject to the same issues as mentioned earlier. Another possibility is to use contrastive learning (Chen et al., 2020; Oord et al., 2018), but without additional constraints, these methods may not distinguish between relevant and irrelevant features.

The ongoing discussion illustrates the fundamental challenge in learning task-relevant features: some objectives (e.g., reconstruction) capture too much information, whereas others (e.g., rewards, inverse models, empowerment) capture too little. Empirically we find that a weighted loss function that combines these objectives does not lead to task-relevant features (see Figure 1). In this work, we revisit feature learning by combining image reconstruction and reward prediction but propose to explicitly "explain away" irrelevant features by constructing a cooperative two-player game between two models. These models, dubbed as task and distractor models, learn task-relevant (s_t^+) and irrelevant features (s_t^-) of the observation (o_t) respectively. Similar to prior work, we force the task model to learn task-relevant features (s_t^+) by predicting the reward. But unlike past work, we also force the distractor 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 implements 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., 2020) formulation in partitioning the state-space into two separate components. However, (Zhang et al., 2020) neither proposes a practical method for segregating relevant information nor provides any experimental validation of their framework 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., 2021) 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 $\mathcal{J}_\pi = \operatorname{argmax}_\pi \mathbb{E} \sum_t \gamma^{t-1} r_t$ discounted by γ.

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

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

Model Learning Dreamer (Hafner et al., 2020) forecasts in a feature representation of images learned via supervision from 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 $\left[\mathcal{J}_{\mathrm{D}}^t \doteq -\beta \mathrm{KL}\big(p(s_t \mid s_{t-1}, a_{t-1}, o_t) \big\| q(s_t \mid s_{t-1}, a_{t-1})\big)\right]$. The overall objective is:

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

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., 2021).

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

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

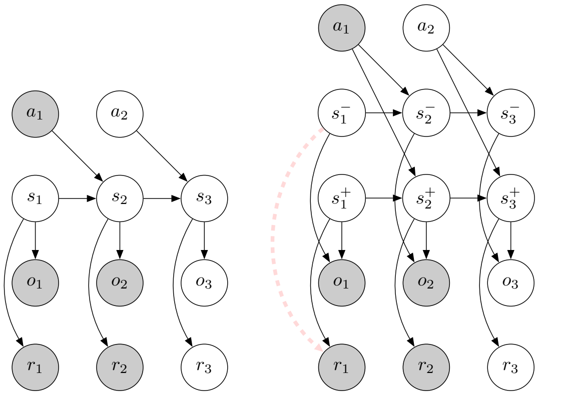
The action model is trained to maximize cumulative rewards over a fixed horizon H. Both the action and value models are learned using imagined rollouts from the learned dynamics. We refer the reader to (Hafner et al., 2020) for more details.

3. Learning Task Informed Abstractions

Task Informed MDP In many real-world problems, the state space of MDP cannot be directly accessed but needs to be inferred from high-dimensional sensory observations. Figure 2a shows the graphical model describing this common scenario. To explicitly segregate task-relevant and irrelevant factors, we propose to model the latent embedding space $\mathcal{S}$ with 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: $\mathrm{MI}(r_t; s_t^-) = 0, \forall t$.

In the most general case, s_{t+1}^- can depend on s_t^+ and s_{t+1}^+ can depend on s_t^-. However, in many realistic scenarios the task-relevant and distractor features evolve independently (e.g. the cars on the road vs leaves flickering in the wind) and thus follow factored dynamics (Guestrin et al., 2003; Pitis et al., 2020). Such a situation greatly simplifies model learning. For this reason we further incorporate this factored structure into our formulation through the assumption: $p(s_{t+1}|s_t, a_t) = p(s_{t+1}^+|s_t^+, a_t)p(s_{t+1}^-|s_t^-, a_t)$.

The resulting MDP, which we call **Task Informed MDP** (TiMDP) is illustrated in Figure 2b. Note that both $\mathcal{S}^+$ and $\mathcal{S}^-$ generate the 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 Section 3.



(a) MDP (b) Task-Informed MDP

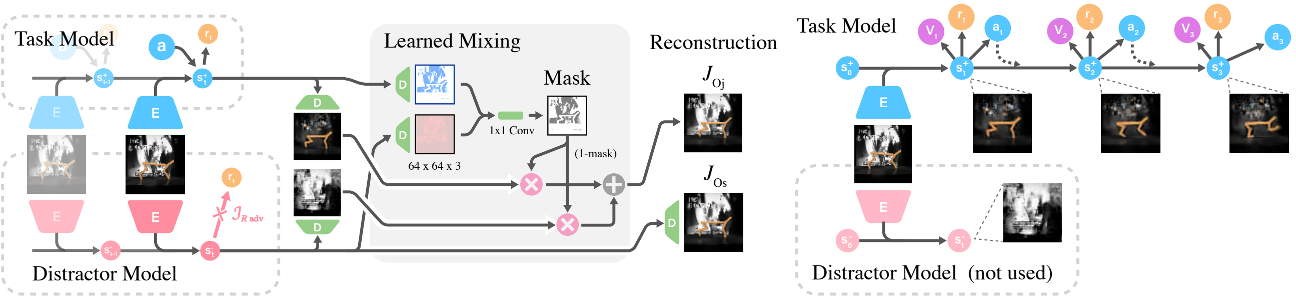
Figure 2: (a) The graphical model of an MDP. (b) Task-Informed MDP (TiMDP). The state space decomposes into two components: s_t^+ captures the task-relevant features, whereas s_t^- captures the task-irrelevant features. The cross-terms between $s^{+/-}$ are removed by imposing a factored MDP assumption.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

Table 1: 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 $s_t^{-/+}$

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

Our method involves learning two models: one model captures the task-relevant state component s_t^+, which we call the *task model*. The other model captures the task-irrelevant state component s_t^-, which we call the *distractor model*. The learning objective for these two models are denoted by $\mathcal{J}_{\mathrm{T}}$ and $\mathcal{J}_{\mathrm{S}}$ (task and distractor), and expanded in Equa-



(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.

tion (4). A visual illustration is provided in Figure 3a. We will explain each component in the following section.

Reward Dissociation for the distractor model is accomplished via the adversarial objective $\mathcal{J}^t_{\text{Radv}}$. This is a minimax setup where we interleave optimizing the distractor model's reward prediction head (for multiple iterations/training step) with the training of the distractor model. While the reward prediction head is trained to minimize the reward prediction loss $-\ln q(r_t|s^-_t)$, the distractor model maximizes this objective so as to exclude reward-correlated information from its learned features (Ganin & Lempitsky, 2015). The reward prediction loss is computed using $\ln \mathcal{N}(r_t; \hat{r}_t, 1)$, where $\mathcal{N}(\cdot; \mu, \sigma^2)$ denotes the Gaussian likelihood, and $\hat{r}_t$ is the predicted reward.

Cooperative Reconstruction By jointly reconstructing the image, the distractor model that's biased towards capturing task-irrelevant information will enable the task model to focus on 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, Hafner et al. 2019) to produce the states $s^+_t$and s^-_t, which are then used 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 with a mask M_t, which we found to be simple and effective.

We produce the mask M_t by letting the two image decoders additionally produce two tensors of size $H \times W \times 3$, where H, W are the height and width of the observed image. These two tensors are concatenated channel-wise to obtain a tensor of size $H \times W \times 6$, then is passed through a 1×1 convolution layer followed by sigmoid activation to obtain the $H \times W \times 1$ mask M_t with 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 $\ln \mathcal{N}(o_t; \hat{o}_t, 1)$. An illustration of the mixing process is in Figure 3a.

Distractor-model-only Reconstruction One failure mode of the formulation being discussed is a degenerate solution where a distractor model that captures no information at all can still satisfy the two objectives described above. This would result in the task model reconstructing the entire observation by itself. To avoid such degeneracy, we add an additional image decoder that encourages the distractor model to reconstruct the entire input observation by itself, so as to capture as much information about the observation as possible. This reconstruction loss is denoted as $\mathcal{J}^t_{Os}$.

Policy Learning is similar to Dreamer, except that we replace the world model with the task model. This way, the forward predictions are only made in the $\mathcal{S}^+$ subspace.

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

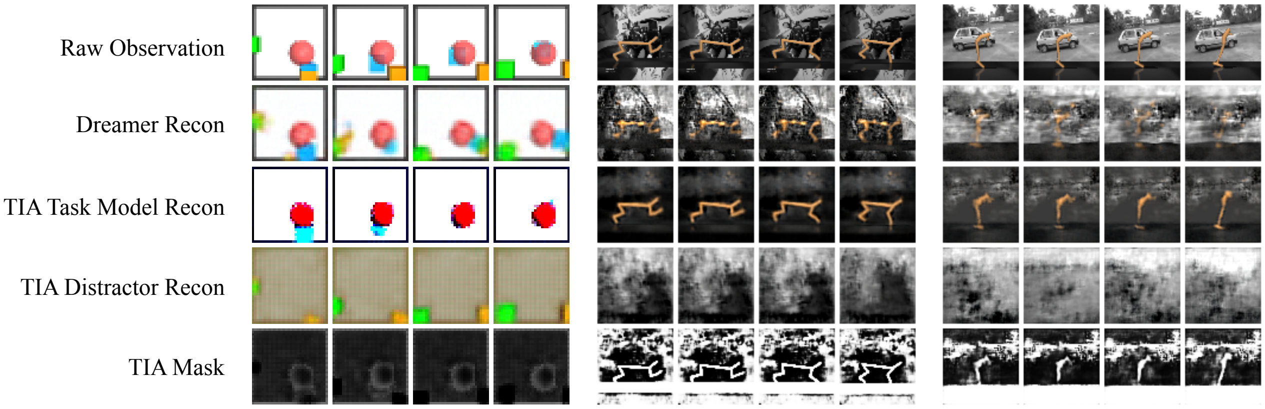
An illustration of the policy learning stage is in Figure 3b.

4. Experiments

Our empirical evaluation aims to answer if the proposed method outperforms existing methods when learning in environments with visually complex backgrounds containing task-irrelevant information. For this purpose, we make use of three environments that are described in Section 4.1. We compare our method against several baselines described in Section 4.2. Our code is available at https://github.com/kyonofx/tia.

4.1. Environments

ManyWorld (Figure 4a) We first experimented on a custom environment suite called ManyWorld where the amount



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

Figure 4: Visualizing the information represented by Dreamer (Hafner et al., 2020) and the task and distractor models of our method on several environments. (a) In the ManyWorld environments, Dreamer mistakes the distractor (yellow) for the target object (blue). The task 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 task model and task-irrelevant information with the distractor model.

of distraction can be easily controlled. The task is to move a target block (in blue) to a goal location. The goal is visually indicated by a translucent red sphere. Other objects are distractors. The amount of distraction can be controlled by varying the number and / or the dynamics of these objects. We turned off physical collisions between objects, but they visually occlude each other if they are present at overlapping positions. The task-relevant objects and the distractor objects are visually similar, necessitating additional effort to resolve when learning a world model. The observation is RGB image of size $32 \times 32 \times 3$.

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., 2021), 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 contains many factors of variation which makes world-model learning challenging and negatively affects downstream policy optimization. The observation is RGB image of size $64 \times 64 \times 3$. The video backgrounds are from the class "driving car" and are grayscale as in Zhang et al. (2021).

Arcade Learning Environments (ALE) or Atari, (Figure 6) is 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 six games. Each seed takes ten days on a V100 Volta GPU. We report 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. The observation is a grayscale image of size $64 \times 64 \times 1$ as in Hafner et al. (2021).

4.2. Baseline Methods

We compare the proposed method against both model-based and model-free baselines. Some of the baseline methods were proposed specifically to tackle learning in the presence of visual distractions. In particular we compare against **Dreamer** (Hafner et al., 2020) 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., 2021). We compare against a strong model-free method, **Deep Bisimulation for Control (DBC)** (Zhang et al., 2021), which uses the bisimulation metric developed specifically to be invariant to task-irrelevant features. Finally, we also include **DeepMDP** (Gelada et al., 2019) which augments the representation with a forward model, then uses model-free, distributional Q learning for the policy. The DeepMDP and DBC results are adapted from Zhang et al. (2021).

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

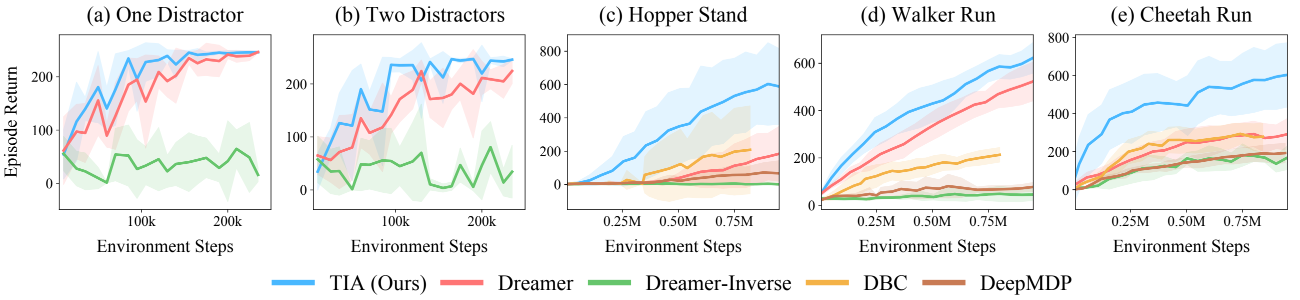


Figure 5: Our method consistently outperforms Dreamer and other baseline methods in a variety of visual control tasks with distraction. The curves show mean and standard deviation, over five seeds for TIA, Dreamer, and Dreamer-Inverse. Results for DBC and DeepMDP are adopted from results reported in (Zhang et al., 2021) 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.

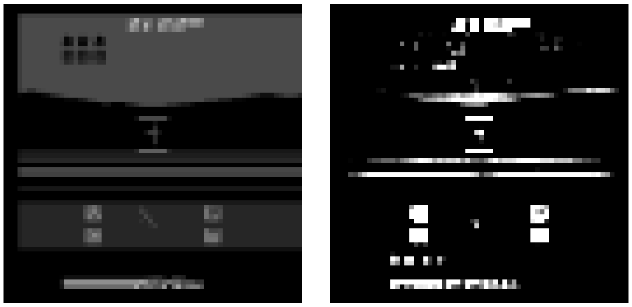


Figure 6: (Left) Raw observation of Atari Robotank. (Right) The task model of TIA emphasizes task-relevant information such as the crosshair and the radar for tracking enemies, while ignoring task-irrelevant information such as textures in the raw observation.

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 objective 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 task model of TIA isolates the agent (blue) and the goal (red), and ignores the rest. On the other hand, the TIA's distractor model successfully captures the distractors (yellow and green). The successful dissociation of task-irrelevant information is confirmed by quantitative results in Figure 5 (a) and (b) showing 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 report the learning curves of *Cheetah Run*, *Walker Run* and *Hopper Stand* are in Figure 5. 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 task 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. Results on additional DMC environments are included in the appendix.

4.4. Results on Atari

In environments tested so far 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 distraction. 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 six Atari games that are known to be challenging for model-based methods (Hafner et al., 2020) 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 sub-

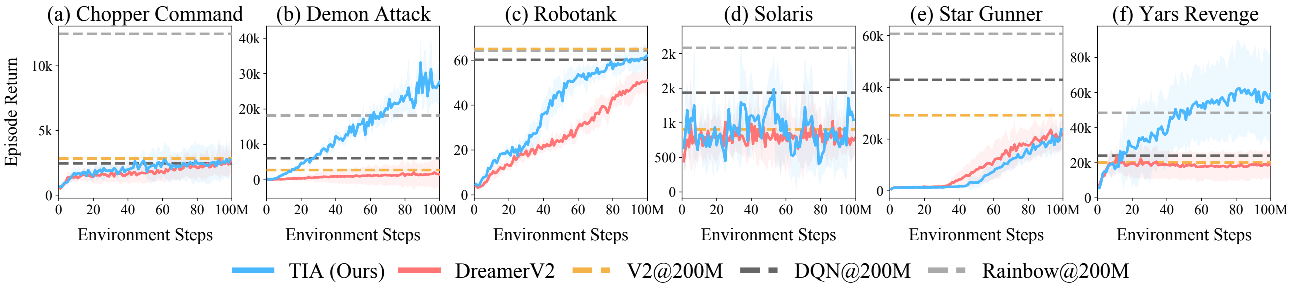


Figure 7: Atari performance at 100M steps. TIA significantly improves policy learning in Demon Attack, Robotank, and Yars Revenge, in which DreamerV2 fails to learn or has inferior sample complexity. We also add the performance of DreamerV2 at 200M steps, along with the performance of two model-free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

stantially outperform the strong baseline of DreamerV2. Furthermore, for the games of *Demon Attack*, *Robotank* and *Yars Revenge*, we match the performance or outperform strong model-free baselines of DQN/Rainbow (Mnih et al., 2013; Hessel et al., 2018) trained for 200M steps, while our method is only trained for 100M steps. These results convincingly demonstrate the superiority of our method in visually cluttered domains. Figure 6 shows the image reconstruction of TIA's task model in *Robotank*.

4.5. Hyperparameter Selection

The two important hyperparameters are λ_{Radv} and λ_{Os}. The balance of these two terms enables the distractor model to capture the task-irrelevant part of states. One particular mode of failure is when the distractor model takes over the reconstruction. It strips the task model from task-relevant information, thus preventing the policy from learning 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 distractor model from dominating. This scenario suggests using a large λ_{Radv} at the beginning of training and slowly increasing the weight λ_{Os} for the distractor reconstruction loss.

On the other extreme the distractor can become degenerate and capture no information of the input observations. In this case, TIA is equivalent to Dreamer. To avoid such degeneracy we would increase λ_{Os} so that the distractor model is encouraged to capture more information.

To gain more insight into whether the reward dissociation process works as expected, we computed the how much reward specific information is captured by the distractor model capture during. For this we recorded the reward prediction error measured using the log-likelihood (as unit Gaussian) in Figure 8. Since the error is unit-less, for the sake of comparison, we estimated the upper bound of the prediction error using the running average of the reward in all historical data. This corresponds to an uninformed reward predictor (the "Mean Predictor") that always guesses the average. The reward prediction loss of the distractor 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 $-\ln \mathcal{N}(0;\ 0,1) = -\ln \frac{1}{\sqrt{2\pi}} \approx 0.92$. This result indicates that the task model learns task-relevant information, while the distractor model's learned feature is indeed reward-independent.

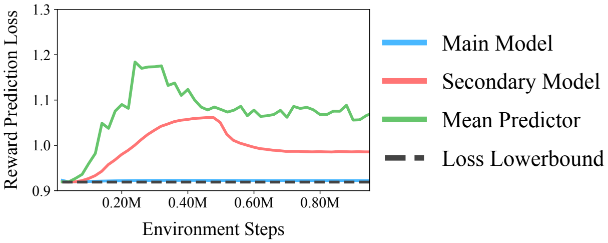


Figure 8: **Reward Dissociation During Learning** We plot the negative log-likelihood loss of the reward prediction $-\ln p(r)$, (lower bounded by 0.92) of a mean predictor, the reward prediction module of the task model, and the reward prediction module of the distractor model. The features from the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

5. Other Related Works

State-of-the-art deep reinforcement learning algorithms often jointly optimize the discounted return together with an auxiliary representation learning objective such as image reconstruction (Watter et al., 2015; Wahlström et al., 2015) or contrastive learning (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., 2017; Tamar et al., 2016; Feinberg et al., 2018; Schrittwieser et al., 2020) 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., 2020a; Yarats et al., 2021; Raileanu et al., 2020; Laskin et al., 2020b; 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. Our approach focuses on finding ways to inform representation learning for RL which features are more useful for the task, and therefore should be learned first.

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) exploit additional structure to learn state abstractions, leading to impressive empirical gains on natural scene domains(Zhang et al., 2021; Gelada et al., 2019; Agarwal et al., 2021). The work presented here falls 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 TIA relies on the task specification for feature separation. It is 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., 2020; 2021; Srinivas et al., 2018; Pertsch et al., 2020).

6. Conclusion and Discussion

This work demonstrates that the TiMDP formulation offers a way for a model-based agent to explain away task-irrelevant information through a distractor model, and 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 two standard benchmarks. TIA reduces to Dreamer when its distractor model degenerates and captures no information. TIA would outperform Dreamer when the hyperparameters are well-tuned and the disentangled state representation would help policy learning.

If we are able to distinguish task-relevant features from irrelevant ones, one might expect that such a scheme would also lead to better transfer to new scenarios. To investigate this, we ran experiments on DMC tasks, where the train and test backgrounds consisted of different videos. We found that when the train and test sets had different sets of natural video backgrounds, our method could easily generalize. However, when we trained on random noise background and test on natural video backgrounds without any fine-tuning, the performance suffered. This result is unsurprising, and the transfer problem can be alleviated by training on diverse data. Further, in our current setup, the learning of task-relevant information only happens during training. However, there is no mechanism to adapt what is relevant/irrelevant at test time. Investigating test-time adaptation methods can improve transfer performance to out-of-distribution situations.

A second issue worth mentioning is that while one set of hyperparameters works well across Atari games, for the DMC environments, the choice of hyperparameters, λ_{Radv} and λ_{Os}, is domain-dependent. We discussed good practices for choosing these hyperparameters in Section 4.5. We hypothesize that based the practice documented here, 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.

It is possible that the factored dynamics assumption in TiMDP is too strong. Take ManyWorld as an example; this assumption is violated when distractor blocks interact with the target and are no longer task-independent. In practice, TIA implements a relaxed version of the factored dynamics constraint via cooperative reconstruction: Features that interact weakly (such as occlusion) can be filtered into the distractor model (see Figure 4a), whereas ones that interact strongly (such as via collision) are highly task-relevant, therefore will be included in the task model.

Our goal in this work was to build agents that operate from complex visual imagery. 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.

Acknowledgement

We would like to thank Amy Zhang for sharing the performance data of DBC and DeepMDP and Danijar Hafner for sharing the performance data of DreamerV2, DQN, and Rainbow. This work is supported by the MIT-IBM grant on adversarial learning of multi-modal and structured data, the MIT-IBM Quest Program, the DARPA Machine Common Sense Program, and the National Science Foundation under Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, http://iaifi.org/). The authors also acknowledge the MIT SuperCloud and the Lincoln Laboratory Supercomputing Center for providing HPC resources.

References

Agarwal, R., Machado, M. C., Castro, P. S., and Bellemare, M. G. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. In *International Conference on Learning Representations*, 2021.

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

Agrawal, P., Nair, A. V., Abbeel, P., Malik, J., and Levine, S. Learning to poke by poking: Experiential learning of intuitive physics. In *Advances in Neural Information Processing Systems*, pp. 5074–5082, 2016.

Bahng, H., Chun, S., Yun, S., Choo, J., and Oh, S. J. Learning de-biased representations with biased representations. In *International Conference on Machine Learning*, pp. 528–539. PMLR, 2020.

Banijamali, E., Shu, R., Bui, H., Ghodsi, A., et al. Robust locally-linear controllable embedding. In *International Conference on Artificial Intelligence and Statistics*, pp. 1751–1759. PMLR, 2018.

Bengio, Y. Deep learning of representations: Looking forward. In Dediu, A., Martín-Vide, C., Mitkov, R., and Truthe, B. (eds.), *Statistical Language and Speech Processing - First International Conference, SLSP 2013, Tarragona, Spain, July 29-31, 2013. Proceedings*, volume 7978 of *Lecture Notes in Computer Science*, pp. 1–37. Springer, 2013. doi: 10.1007/978-3-642-39593-2_1.

Cadene, R., Dancette, C., Cord, M., Parikh, D., et al. Rubi: Reducing unimodal biases for visual question answering. In *Advances in neural information processing systems*, pp. 841–852, 2019.

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.

Clark, C., Yatskar, M., and Zettlemoyer, L. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In Inui, K., Jiang, J., Ng, V., and Wan, X. (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.

Clark, C., Yatskar, M., and Zettlemoyer, L. Learning to model and ignore dataset bias with mixed capacity ensembles. In Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pp. 3031–3045. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.272.

Dean, T. L., Givan, R., and Leach, S. M. Model reduction techniques for computing approximately optimal solutions for markov decision processes. In Geiger, D. and Shenoy, P. P. (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.

Du, S., Krishnamurthy, A., Jiang, N., Agarwal, A., Dudik, M., and Langford, J. Provably efficient rl with rich observations via latent state decoding. In *International Conference on Machine Learning*, pp. 1665–1674. PMLR, 2019.

Feinberg, V., Wan, A., Stoica, I., Jordan, M. I., Gonzalez, J. E., and Levine, S. Model-based value estimation for efficient model-free reinforcement learning. *arXiv preprint arXiv:1803.00101*, 2018.

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

Ferns, N., Panangaden, P., and Precup, D. Metrics for finite markov decision processes. In *UAI*, volume 4, pp. 162–169, 2004.

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

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

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.

Gelada, C., Kumar, S., Buckman, J., Nachum, O., and Bellemare, M. G. Deepmdp: Learning continuous latent space models for representation learning. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2170–2179. PMLR, 2019.

Givan, R., Dean, T. L., and Greig, M. 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.

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

Guestrin, C., Koller, D., Parr, R., and Venkataraman, S. Efficient solution algorithms for factored mdps. *Journal of Artificial Intelligence Research*, 19:399–468, 2003.

Ha, D. and Schmidhuber, J. Recurrent world models facilitate policy evolution. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 2455–2467, 2018.

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

Hafner, D., Lillicrap, T. P., Ba, J., and Norouzi, M. 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, 2020.

Hafner, D., Lillicrap, T. P., Norouzi, M., and Ba, J. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

He, H., Zha, S., and Wang, H. Unlearn dataset bias in natural language inference by fitting the residual. In Cherry, C., Durrett, G., Foster, G. F., Haffari, R., Khadivi, S., Peng, N., Ren, X., and Swayamdipta, S. (eds.), *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP, DeepLo@EMNLP-IJCNLP 2019, Hong Kong, China, November 3, 2019*, pp. 132–142. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-6115.

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

Higgins, I., Pal, A., Rusu, A., Matthey, L., Burgess, C., Pritzel, A., Botvinick, M., Blundell, C., and Lerchner, A. Darla: Improving zero-shot transfer in reinforcement learning. In *International Conference on Machine Learning*, pp. 1480–1490. PMLR, 2017.

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

Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., Mohiuddin, A., Sepassi, R., Tucker, G., and Michalewski, H. 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.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

Kingma, D. P. and Welling, M. Stochastic gradient vb and the variational auto-encoder. In *Second International Conference on Learning Representations, ICLR*, volume 19, 2014.

Kingma, D. P., Mohamed, S., Jimenez Rezende, D., and Welling, M. Semi-supervised learning with deep generative models. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 3581–3589. Curran Associates, Inc., 2014.

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

Laskin, M., Lee, K., Stooke, A., Pinto, L., Abbeel, P., and Srinivas, A. Reinforcement learning with augmented data. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020a.

Laskin, M., Srinivas, A., and Abbeel, P. CURL: contrastive unsupervised representations for reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5639–5650. PMLR, 2020b.

Lee, K., Fischer, I., Liu, A., Guo, Y., Lee, H., Canny, J., and Guadarrama, S. Predictive information accelerates learning in RL. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Mazoure, B., des Combes, R. T., Doan, T., Bachman, P., and Hjelm, R. D. Deep reinforcement and infomax learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

McCallum, R. Reinforcement learning with selective perception and hidden state. 1997.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Oh, J., Singh, S., and Lee, H. Value prediction network. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6118–6128, 2017.

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

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

Pertsch, K., Rybkin, O., Ebert, F., Zhou, S., Jayaraman, D., Finn, C., and Levine, S. Long-horizon visual planning with goal-conditioned hierarchical predictors. *Advances in Neural Information Processing Systems*, 33, 2020.

Pitis, S., Creager, E., and Garg, A. Counterfactual data augmentation using locally factored dynamics. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Racanière, S., Weber, T., Reichert, D. P., Buesing, L., Guez, A., Rezende, D., Badia, A. P., Vinyals, O., Heess, N., Li, Y., 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.

Raileanu, R., Goldstein, M., Yarats, D., Kostrikov, I., and Fergus, R. Automatic data augmentation for generalization in deep reinforcement learning. *arXiv preprint arXiv:2006.12862*, 2020.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609, 2020.

Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., Levine, S., and Brain, G. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1134–1141. IEEE, 2018.

Silver, D., van Hasselt, H., Hessel, M., Schaul, T., Guez, A., Harley, T., Dulac-Arnold, G., Reichert, D. P., Rabinowitz, N. C., Barreto, A., and Degris, T. The predictron: End-to-end learning and planning. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3191–3199. PMLR, 2017.

Srinivas, A., Jabri, A., Abbeel, P., Levine, S., and Finn, C. Universal planning networks: Learning generalizable representations for visuomotor control. In *International Conference on Machine Learning*, pp. 4732–4741. PMLR, 2018.

Stone, A., Ramirez, O., Konolige, K., and Jonschkowski, R. The distracting control suite–a challenging benchmark for reinforcement learning from pixels. *arXiv preprint arXiv:2101.02722*, 2021.

Stooke, A., Lee, K., Abbeel, P., and Laskin, M. Decoupling representation learning from reinforcement learning. *arXiv preprint arXiv:2009.08319*, 2020.

Tamar, A., Levine, S., Abbeel, P., Wu, Y., and Thomas, G. Value iteration networks. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 2146–2154, 2016.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., de Las Casas, D., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., Lillicrap, T., and Riedmiller, M. Deepmind control suite. January 2018.

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

Wang, H., Ge, S., Lipton, Z., and Xing, E. P. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pp. 10506–10518, 2019.

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

Yan, W., Vangipuram, A., Abbeel, P., and Pinto, L. Learning predictive representations for deformable objects using contrastive estimation. *arXiv preprint arXiv:2003.05436*, 2020.

Yarats, D., Zhang, A., Kostrikov, I., Amos, B., Pineau, J., and Fergus, R. Improving sample efficiency in model-free reinforcement learning from images. October 2019.

Yarats, D., Kostrikov, I., and Fergus, R. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2021.

Zhang, A., Lyle, C., Sodhani, S., Filos, A., Kwiatkowska, M., Pineau, J., Gal, Y., and Precup, D. Invariant causal prediction for block mdps. In *International Conference on Machine Learning*, pp. 11214–11224. PMLR, 2020.

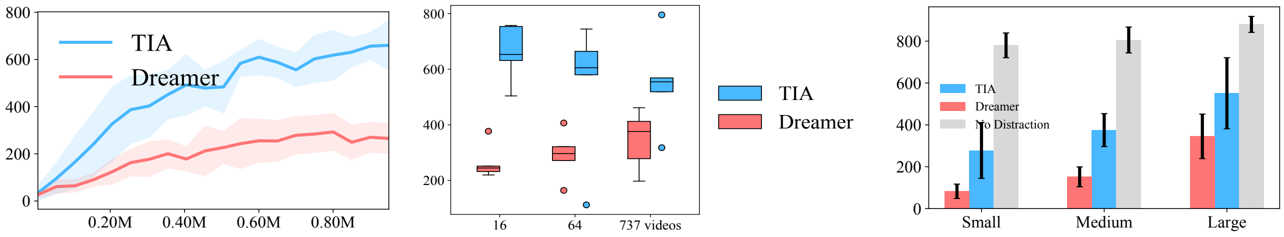
Zhang, A., McAllister, R. T., Calandra, R., Gal, Y., and Levine, S. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021.

A. Experimental Details

A.1. Additional Ablation

The learning curves for the DMC domains in Figure 5 has large variance. We run a two-tailed test for performance at 1M steps, the results are: $[t=3.17,\ p=0.014]$, $[t=2.28,\ p=0.052]$, and $[t=2.21,\ p=0.061]$ on *Hopper*, *Walker* and *Cheetah*, respectively. A major source of variance in the result is that the set of 737 videos in the natural video dataset are only sampled 1000 times in total (at the beginning of each episode). Therefore the model is constantly surprised by visually distinct videos (see outliers in Figure 9b), leading to large variance in the episodic reward for all algorithms considered. To control for this, we present additional results on *Cheetah Run* that only uses 16 videos in Figure 9a and a comparison across $16, 64, 737$ videos in a box plot (Figure 9b). The variance decreases significantly when this is controlled for, showing that TIA outperforms Dreamer. One interesting observation is that Dreamer performs worse with a smaller number of videos (potentially because it prioritizes memorization of the video background).

We further reproduce Figure 1 while including TIA's performance in Figure 9c. TIA significantly outperforms Dreamer irrespective of the model size.



(a) With 16 videos (b) Variance from number of videos (c) Varying model size

Figure 9: (a) With 16 videos used in the environment, the learning curve has much smaller variance, showing TIA significantly outperforms Dreamer. (b) Performance at 1M steps over the five seeds, using different number of videos in the environment. (c) TIA outperforms Dreamer irrespective of the model size at 1M steps.

A.2. 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 10, where we include seven additional (ten in total) domains from the DeepMind Control suite. In *Finger Spin* the gap almost disappears due to poor performance of the model-based agent in the clean environment – 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 selected domains as a proxy for how much visual distraction is introduced: Walker 63%, Hopper 77%, Cheetah: 83%.

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

A.3. Bridging the Gap by Learning Task Informed Abstractions

We produce the complete result on DeepMind Control Suite, including the seven additional tasks from above. We set up the experiments by matching the total number of parameters so that each one of TIA's two models is only half in size as the single world model from Dreamer. This comparison is a disadvantage to our method because the smaller task model runs the risk of being too small to capture the set of task-relevant features in its entirety.

Despite of this disadvantage in model capacity, TIA is able to reduce the gap on *Walker Run*, *Hopper Hop*, and *Reacher Easy* while making the gap significantly smaller on *Hopper Stand* and *Cheetah Run* (see Figure 10). On *Walker Stand* and *Walker Walk*, the gap is small to begin with. Therefore our method cannot bring much benefit. Domains such as *Cartpole Swingup* and *Cup Catch* are devastatingly hard for Dreamer to master under visual distraction due to the partial-observable/small object manipulation nature of the domains. To conclude, since our implementation of TIA is a Dreamer with an additional capability of ignoring task-irrelevant features, it would not perform worse than Dreamer. However, tasks that are inherently hard for model-based methods would remain hard for TIA.

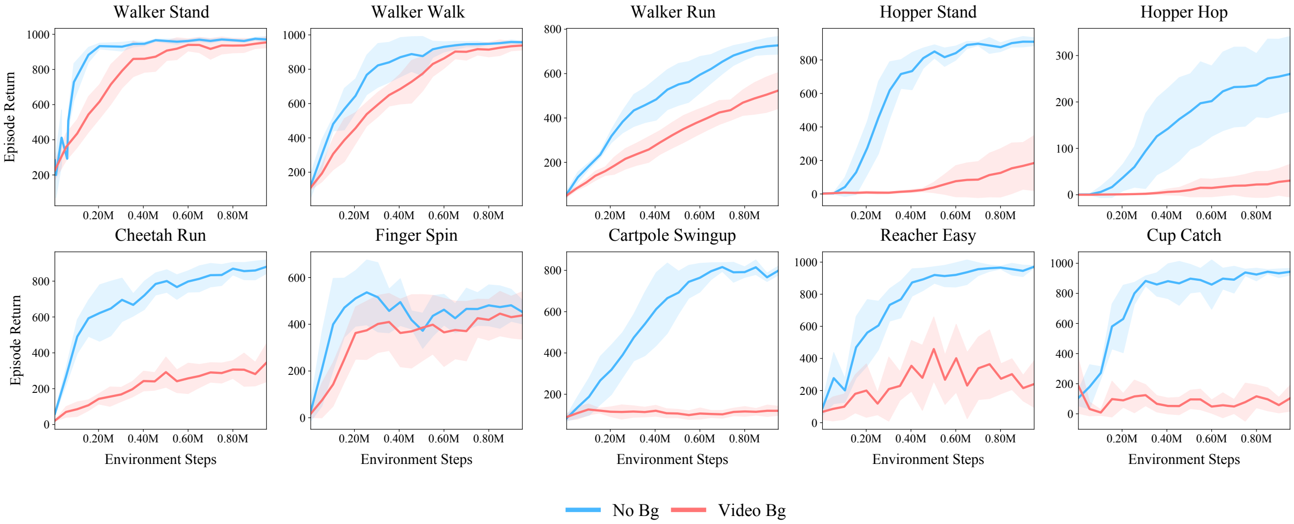


Figure 10: **The data efficiency and performance gap** when learning with video distraction backgrounds.

A.4. Understanding Failure Cases

In Section 4.5, we provide the principled approach to tuning the hyperparameters 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 distractor model or the task model takes over, as type I and type II. We provide detailed renderings of these failure cases below Figure 12 in comparison to a successfully learned world model.

The first column in Figure 12a shows a successfully trained agent whose task model is able to perfectly reconstruct the walker agent. In type I failure mode (Figure 12b), the distractor model would take over the entire reconstruction, causing the task 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 $\mathcal{J}_{\mathrm{Radv}}$ 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 distractor model from the reward, or decrease λ_{Os} that weighs the distractor reconstruction term.

In the type II failure case (see Figure 12c), the task 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 distractor 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.

B. Model Architecture Details

Model Detals and Sizes For fair comparisons, 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 distractor 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., 2020).

On the Arcade Learning Environments, we compare against the state-of-the-art on this domain, DreamerV2 (Hafner et al., 2021), which uses a world model of 20 million trainable parameters. We made the task model the same size as the original

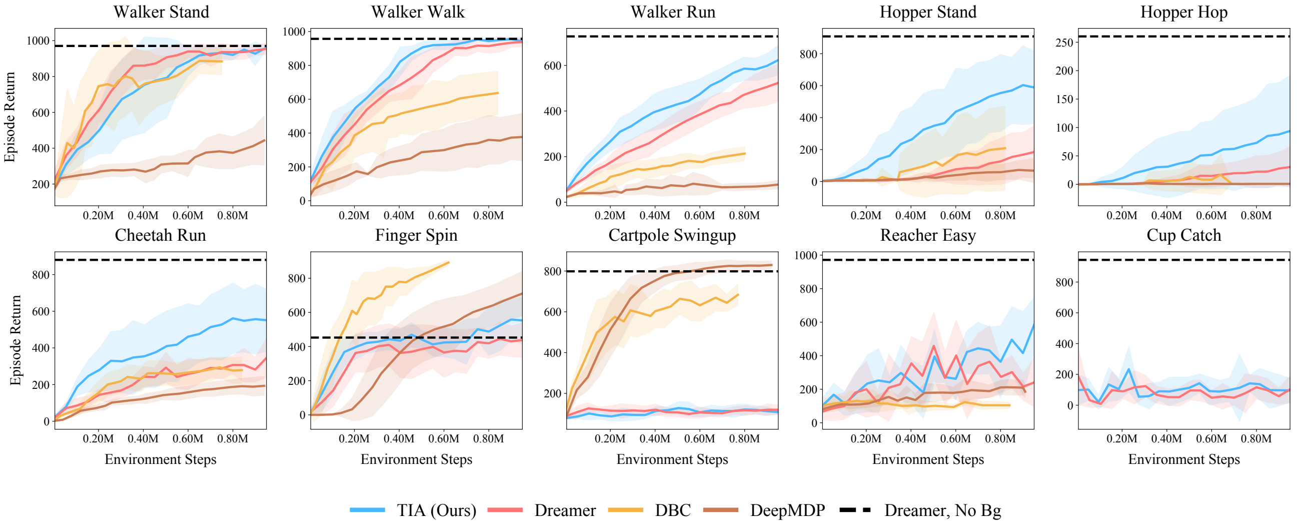


Figure 11: **Performance of Task Informed Abstractions** on ten DeepMind control domains. Zhang et al. 2021 did not evaluate DBC/DeepMDP on *Cup Catch* so we only report results of TIA and Dreamer for *Cup Catch*.

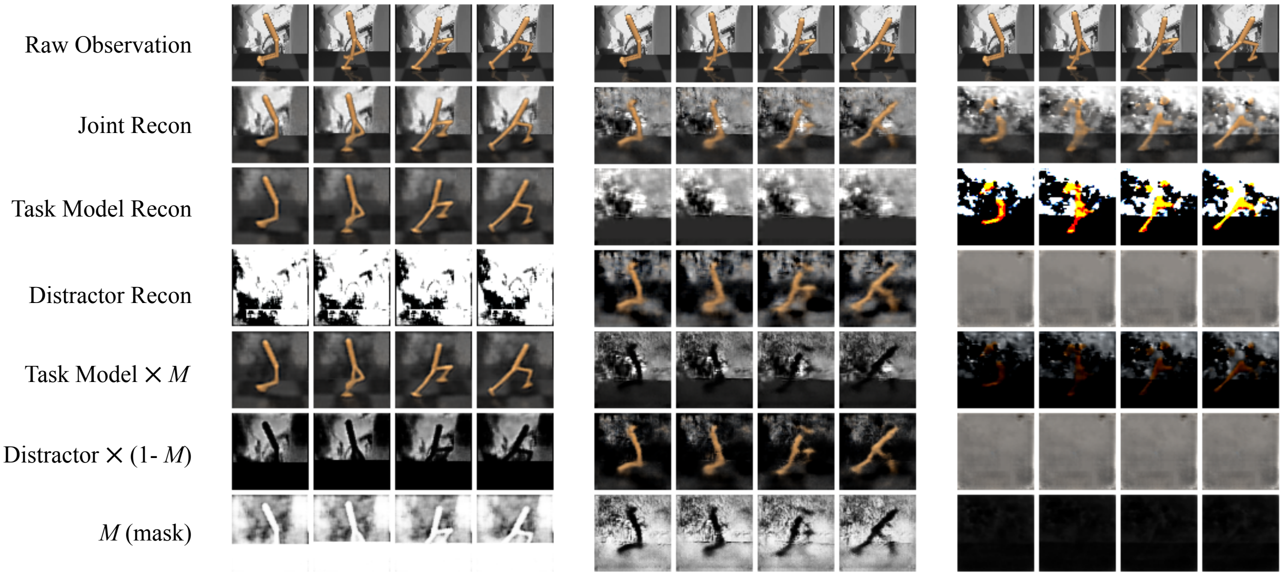
implementation while adding a smaller distractor model which contains 12 million parameters. A key difference between Dreamer and DreamerV2 is that the latter has an additional prediction head for the discount factor γ_t besides the standard reward prediction head. This discount factor head plays an instrumental role in allowing DreamerV2 to solve Atari games. Therefore we additionally dissociate the distractor 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., 2021).

We use an input size of $32 \times 32 \times 3$ in ManyWorld, $64 \times 64 \times 3$ in DeepMind Control Suite, and $64 \times 64 \times 1$ in Atari games. We use grayscale for the natural video backgrounds, the same as previous work (Zhang et al., 2021).

Hyperparameters For fair comparisons, we did not tweak existing hyperparameters from Dreamer and used identical settings as Hafner et al. (2020) and Hafner et al. (2021). 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 2 mostly reflect the differences in input image sizes. We tweaked λ_{Os} to stabilize training. Detailed settings for each domain are in Table 2.

Table 2: Hyperparameters

Domain and Task	λ_{Radv}	λ_{Os}
ManyWorld, 1 Distractor	600.0	2.0
ManyWorld, 2 Distractor	150.0	2.0
Cartpole Swingup	30k	2.0
Hopper Stand	30k	2.0
Hopper Hop	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
Reacher Easy	20k	1.0
All ATARI games	2k	1.0



(a) Success

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

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

Figure 12: Detailed rendering from one successfully trained walker agent, and two failure cases of type 1 and type 2. In type 1 failure case, the distractor model takes over the entire reconstruction, rendering the task model ineffective for policy learning (random policy). In the type 2 failure case, the task 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.

C. 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 3 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 3).

Table 3: **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 a 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 4).

Table 4: **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, a mechanism to reject unseen backgrounds at decision time is required to transfer successfully. This points to the incorporation of attention as a potential avenue for future work.