Overcoming Knowledge Barriers: Online Imitation Learning from Observation with Pretrained World Models

Xingyuan Zhang [1 2] **Philip Becker-Ehmck** [1] **Patrick van der Smagt** [1 3] **Maximilian Karl** [1]

Abstract

Pretraining and finetuning models has become increasingly popular. But there are still serious impediments in Imitation Learning from Observation (ILfO) with pretrained models. This study identifies two primary obstacles: the Embodiment Knowledge Barrier (EKB) and the Demonstration Knowledge Barrier (DKB). The EKB emerges due to the pretrained models' limitations in handling novel observations, which leads to inaccurate action inference. Conversely, the DKB stems from the reliance on limited demonstration datasets, restricting the model's adaptability across diverse scenarios. We propose separate solutions to overcome each barrier and apply them to Action Inference by Maximising Evidence (AIME), a state-of-the-art algorithm. This new algorithm, AIME-NoB, integrates online interactions and a data-driven regulariser to mitigate the EKB. Additionally, it uses a surrogate reward function to broaden the policy's applicability, addressing the DKB. Our experiments on tasks from the DeepMind Control Suite and Meta-World benchmarks show that AIME-NoB significantly enhances sample efficiency and performance, presenting a robust framework for overcoming the challenges in ILfO with pretrained models.

1. Introduction

We have been going through a paradigm shift from learning from scratch to pretraining and finetuning, in particular in Computer Vision (CV) (He et al., b; Radford et al., a; He et al., a) and Natural Language Processing (NLP) (Devlin et al.; Radford et al., b; Ouyang et al.; Touvron et al., a;b) fields due to the increasing availability of foundation models (Bommasani et al.) and ever-growing datasets. However, it is still unclear how to adapt this new paradigm into decision-making, in particular what type of models we need to pretrain and how these models can be adapted to solve downstream tasks. Recent work (Zhang et al.; DeMoss et al.; Sekar et al.; Rajeswar et al.; Hansen et al., a) showed that pretrained latent space world models enable successful and efficient transfer to new tasks with either reinforcement learning (Sekar et al.; Rajeswar et al.; Hansen et al., a) or Imitation Learning from Observation (ILfO) (Zhang et al.; DeMoss et al.). ILfO (Torabi et al., a;b; Baker et al.; Zhang et al.; DeMoss et al.; Liu et al., a), especially from videos (Baker et al.; Zhang et al.; Liu et al., a; DeMoss et al.), is a more promising approach in this new paradigm since it does not require a handcrafted reward function which is hard to define for many real-world tasks.

But there are challenges when using pretrained models in ILfO. To quantify these we introduce two new barriers, which we call the Embodiment Knowledge Barrier (EKB) and the Demonstration Knowledge Barrier (DKB). The EKB describes the shortcomings of a pretrained model when confronted with novel observations and actions beyond its training experience. The DKB describes the generalisation from a limited number of expert demonstrations in imitation learning (Ho & Ermon). Approaches like BCO(0) (Torabi et al., a) and AIME (Zhang et al.) typically suffer from these two knowledge barriers. First, these algorithms depend on the pretrained model to infer missing actions from observation sequences. Thus, when the model has not seen a specific observation before, it may not know enough about the embodiment to infer the correct action. Second, if the policy optimisation is only guided by limited demonstrations can lead to a policy that generalises poorly, working well in some scenarios but not in others.

To better showcase the two barriers, in Figure 1 left, we evaluate both AIME (Zhang et al.) and BCO(0) (Torabi et al., a) and their oracle versions w.r.t. different number of demonstrations on walker-run task. Both algorithms pretrain a model from a large embodiment dataset and use that to infer the actions for the observation-only demonstrations. The oracle versions remove the need to infer the missing actions, thus removing the EKB. As we can see from the

[*]Equal contribution [1]Machine Learning Research Lab, Volkswagen Group [2]Technical University of Munich [3]Eötvös Loránd University Budapest. Correspondence to: Xingyuan Zhang <xingyuan.zhang@volkswagen.de>.

First Workshop on Controllable Video Generation at ICML 2024, Vienna, Austria. PMLR 235, 2024.

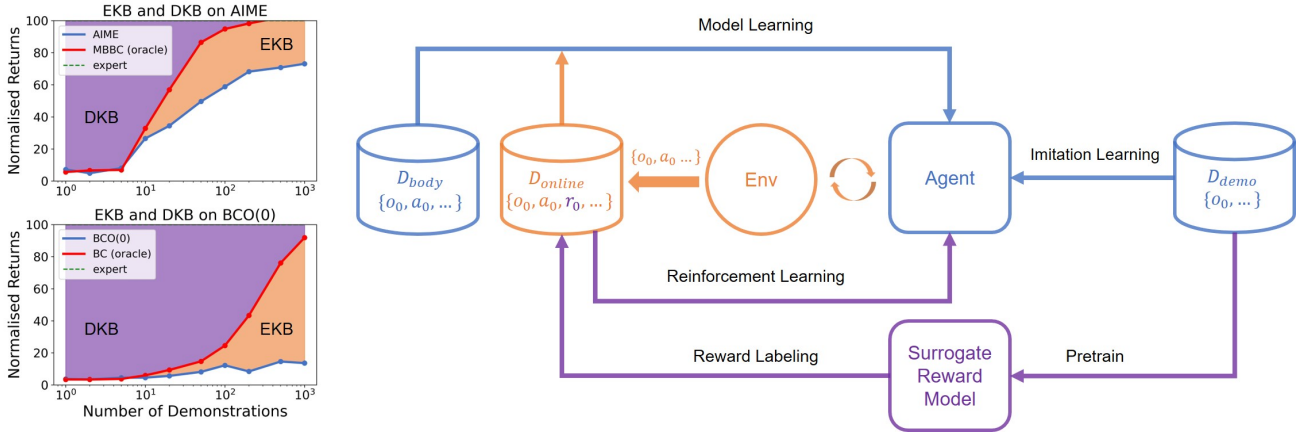


Figure 1. Main idea of this paper. On the left, we plot the performance of BCO(0) and AIME together with their oracle versions w.r.t. different number of demonstrations on walker-run task. The purple region between the oracle version and the expert is the Demonstration Knowledge Barrier (DKB) while the orange region between the algorithm and its oracle version represents the Embodiment Knowledge Barrier (EKB). On the right, we present the solutions proposed in this paper to overcome the two barriers. The blue parts represent the original version of the algorithms that suffer from the knowledge barriers. Orange parts demonstrate the solution for EKB, where the agent is allowed to interact with the environment and use D_{online} together with D_{body} to update the world model. Purple parts show the solution for DKB, where a surrogate reward model is pretrained and used to label the online dataset D_{online} and then used as an RL signal for policy learning.

figure, the two algorithms are always upper-bounded by the corresponding oracle version, and the difference between them represents the EKB. On the other hand, even if given the true actions of the expert, imitation performance may still be impacted by a limited number of demonstrations providing insufficient coverage of the state space. Thus, the difference between the oracle version and the expert performance represents the DKB.

In this paper, we study how to resolve these barriers to improve the performance of ILfO approaches, in particular of AIME. For the EKB, we extend the setting from offline to online by allowing the agent to further interact with the environment to gather more data to train the world model. While for the DKB, we introduce a surrogate reward function to allow the policy to essentially train on more data. We demonstrate that the proposed modifications significantly improve the performance on nine tasks in DeepMind Control Suite (DMC) (Tunyasuvunakool et al.) and six tasks in Meta-World (Yu et al.).

We summarise our contributions as follows:

- We identify and thoroughly analyse the two knowledge barriers, namely EKB and DKB, in the current pretrained-model-based ILfO methods.
- We propose AIME-NoB as an extension of the state-of-the art AIME algorithm by resolving the two knowledge barriers. Specifically, AIME-NoB uses online interaction with data-driven regularisation to solve the EKB and learn a surrogate reward function enlarging state coverage to solve the DKB.
- We evaluate AIME-NoB on 15 tasks from two benchmarks and the results demonstrate AIME-NoB significantly outperforms previous state-of-the-art methods both in terms of final performance and sample efficiency. We also show how the EKB and the DKB are resolved by the proposed modifications with ablation studies.

2. Preliminary

We mostly follow the problem setup as described in Zhang et al.. We consider a POMDP problem defined by the tuple $\{S, A, T, R, O, \Omega\}$, where S is the state space, A is the action space, $T : S \times A \to S$ is the dynamic function, $R : S \to \mathbb{R}$ is the reward function, O is the observation space, which is image in this paper, and $\Omega : S \to O$ is the emission function. The goal is to find a policy $\pi : S \to A$ which maximises the accumulated reward, i.e. $\sum_t r_t$.

We presume the existence of three datasets of the same embodiment available to our agent. The *embodiment dataset* D_{body} contains trajectories $\{o_0, a_0, o_1, a_1 \dots\}$ that represent past experiences of interacting with the environment. This dataset provides information about the embodiment for the algorithm to learn a world model. In addition, we also allow the agent to interact with the environment to collect new data in a *replay buffer* D_{online}. Note that, although the

simulator will give us the reward information, the agent is not allowed to use them, and we only use the reward for evaluation purpose. The *demonstration dataset* D_{demo} contains a few expert trajectories $\{o_0, o_1, o_2 \dots\}$ of the embodiment solving a certain task defined by R_{demo}. The crucial difference between this dataset and the other two datasets is that the actions are not provided anymore since they are not observable from a third-person perspective. The goal of our agent is to learn a policy π from D_{demo} which can solve the task defined by R_{demo} as well as the expert who generated D_{demo}.

2.1. World Models

A World Model (Ha & Schmidhuber) is a generative model which models a probability distribution over sequences of observations, i.e. $p(o_{1:T})$. The model can be either unconditioned or conditioned on other factors, such as previous observations or actions. When the actions taken are known, they can be considered as the condition, i.e. $p(o_{1:T}|a_{0:T-1})$, and the model is called embodied (Zhang et al.).

In this paper, we consider variational latent world models where the observation is governed by a Markovian hidden state. In the literature, this type of model is also referred to as a State-Space Model (SSM) (Karl et al.; Hafner et al., b;a; Becker-Ehmck et al.; Klushyn et al.). Such a variational latent world model involves four components, namely

$$\begin{aligned} \text{encoder } z_t &= f_\phi(o_t), \\ \text{posterior } s_t &\sim q_\phi(s_t|s_{t-1}, a_{t-1}, z_t), \\ \text{prior } s_t &\sim p_\theta(s_t|s_{t-1}, a_{t-1}), \\ \text{decoder } o_t &\sim p_\theta(o_t|s_t). \end{aligned}$$

$f_\phi(o_t)$ is the encoder to extract the features from the observation; $q_\phi(s_t|s_{t-1}, a_{t-1}, z_t)$ and $p_\theta(s_t|s_{t-1}, a_{t-1})$ are the posterior and the prior of the latent state variable; while $p_\theta(o_t|s_t)$ is the decoder that decodes the observation distribution from the state. ϕ and θ represent the parameters of the inference model and the generative model respectively.

Typically, the model is trained by maximising the Evidence Lower Bound (ELBO) which is a lower bound of the log-likelihood, or evidence, of the observation sequence, i.e. $\log p_\theta(o_{1:T}|a_{0:T-1})$. Given a sequence of observations, actions, and states, the objective function is

$$\text{ELBO} = \sum_{t=1}^{T} J_t^{\text{rec}} - J_t^{\text{KL}}, \tag{1}$$

$$\text{where } J_t^{\text{rec}} = \log p_\theta(o_t|s_t), \tag{2}$$

$$J_t^{\text{KL}} = D_{\text{KL}}[q_\phi||p_\theta]. \tag{3}$$

The objective function is composed of two terms: the first term J^{rec} is the likelihood of the observation under the inferred state, which is usually called the reconstruction loss; while the second term J^{KL} is the KL divergence between the posterior and the prior distributions of the latent state. To compute the objective function, we use the re-parameterisation trick (Kingma & Welling; Rezende et al.) to autoregressively sample the inferred states from the observation and action sequence.

In summary, a world model is trained by solving the optimisation problem as

$$\phi^*, \theta^* = \underset{\phi,\theta}{\text{argmax}}\, \mathbb{E}_{\{o,a\}\sim D_{\text{body}}, s\sim q_\phi}[\text{ELBO}]. \tag{4}$$

2.2. AIME

AIME is a state-of-the-art algorithm that uses a pretrained world model to solve ILfO in an offline setting. Specifically, it uses the pretrained world model as an implicit inference model by solving for the best action sequence that makes the demonstration most likely under the trained world model. The imitation can be done jointly with the action inference using amortised inference and the re-parameterisation trick by solving the following optimisation problem

$$\psi^* = \underset{\psi}{\text{argmax}}\, \mathbb{E}_{o\sim D_{\text{demo}}, s\sim q_{\phi^*,\theta^*}, a\sim\pi_\psi}[\text{ELBO}], \tag{5}$$

where ψ is the parameter for policy $\pi_\psi(a_t|s_t)$. The resulting objective is very similar to Equation (4), with a subtle difference of the sampling path. That is in the new objective, only the observations are sampled from the dataset and both states and actions are sampled iteratively from the learned model and the policy, respectively.

3. Methodology

In this section we will analyse the EKB and DKB for AIME. Based on the analysis we introduce a solution for each knowledge barrier and combine them into AIME-NoB, where NoB stands for **No B**arriers. The general framework of the solutions is shown in Figure 1 and the pseudocode of AIME-NoB is in Algorithm 1 in Appendix A.

3.1. Resolving the EKB

The most natural way to solve the EKB is to allow the agent to further interact with the environment. New experiences can minimise the error in the pretrained model in proximity of the policy π_ψ and gain more embodiment knowledge relevant for the task at hand. Torabi et al. (a) proposed a modified version of BCO(0) called BCO(α) which introduced such an interaction phase. However, from their and our empirical results, it did not resolve the EKB since there remains a gap between BCO(α) and the BC oracle when the environment is complex. In fact, as we will show in the following, the idea of adding online interactions is not straightforward to successfully implement in practice.

As shown in recent works in Offline RL, continuing training an actor-critic from the offline phase in the online phase requires certain measures to combat the shift of objective (Lee et al.; Ball et al., 2023; Nakamoto et al.). A similar story also applies when extending AIME from purely offline to online. The most dominant problem we found is overfitting to the newly collected dataset.

As the training progresses iteratively between data collection, model training and policy training, in the early phase of training there are only a few new trajectories available for training the model. Because the world model is highly expressive, it may overly favour similar trajectories in the new data, leading to a high ELBO. Normally, this may not be a big problem since, eventually, more and more data will be collected to combat this overfitting. But since AIME also depends on the ELBO to train the policy, it quickly causes the policy training to diverge.

In order to address the overfitting issue, we need a regulariser for model learning. Instead of designing ad-hoc methods to regularise the model in the parameter space, we adopt a data-driven approach. From the model's perspective, the overfitting is caused by a sudden shift of the training data from a large and diverse pretraining dataset to a small and narrow replay buffer. So one way to make the shift not as sudden is to append the pretraining dataset to the replay buffer, so that the distribution of the training data will change smoothly. However, this causes data efficiency problems since the replay buffer is relatively small compared to the pretraining dataset so that uniformly sampling from them together limits using the new data. Instead, we consider sampling separately from both datasets. We modify the objective in Equation (4) to

$$\begin{aligned}\phi^*, \theta^* = \underset{\phi,\theta}{\operatorname{argmax}}\ & \alpha \mathbb{E}_{\{o,a\}\sim D_{\text{body}}, s\sim q_\phi}[\text{ELBO}] \\ & + (1-\alpha)\mathbb{E}_{\{o,a\}\sim D_{\text{online}}, s\sim q_\phi}[\text{ELBO}]. \end{aligned} \tag{6}$$

The amount of data we sample from the pretraining dataset is controlled by a hyper-parameter α, which represents how much regularisation we put upon the model. Here we mainly consider setting $\alpha = 0.5$, so that we sample the data evenly from both datasets. We justify our choice in the ablation section with Figure 4b.

This finding contradicts Rajeswar et al. and Hansen et al. (a), where the pretrained world models do not need such a data-driven regulariser. We conjecture that unlike AIME, these approaches mainly use their world models purely as generative models to predict states and rewards given action sequences, which is only indirectly influenced by overfitting the ELBO.

3.2. Resolving the DKB

Based on the discussion from the previous sections, the straightforward way of solving the DKB is also to increase the number of demonstrations available to the agent. However, expert demonstrations are difficult and expensive to collect. Increasing the size of the demonstration dataset is not always feasible in real-world applications. In order to propose a more practical solution, we need to look deeper into what is the real cause of the DKB.

The policy-learning part of the AIME algorithm is essentially behaviour cloning, and it is only conducted on the demonstration dataset. So for the states covered in the demonstration dataset, the policy is given clear guidance about what to do, while for other states, the behaviour is undefined. AIME solely relies on the generalisation abilities of the learned latent state and the trained policy network to extrapolate the correct behaviour. In particular for small demonstration datasets, this can be unreliable or even impossible. Therefore, if we can enlarge the space of the covered states, we should reduce the DKB (Ross et al.).

Generative Adversarial Imitation Learning (GAIL) style algorithms (Ho & Ermon; Peng et al.; Torabi et al., b; Liu et al., a) are examples of this solution. Instead of directly learning from the demonstration, they adversarially train a discriminator to assess how closely each state matches the demonstration dataset. When learning the policy, they treat the discriminator's output as a reward and encourage visiting states that are more likely to belong to the demonstrations. This modification provides guidance for newly visited states in the replay buffer, allowing the space of supported states to grow over the course of training.

However, this benefit is not clearly separable in GAIL since it is always entangled with the adversarial training of the discriminator. A recent work MAHALO (Li et al.) shows further evidence of the importance of the size of the covered space. The authors studied a similar ILfO setup with embodiment and demonstration datasets. They compared two variants: for one they train an inverse dynamics model (IDM) from the embodiment dataset and use it to label the demonstration dataset, while for another they train a reward model from the demonstration dataset by labelling all time steps with a reward of 1, and then use it to label the embodiment dataset. Finally, they run the same offline RL algorithm on both labelled datasets. The results show the second variant attains a much better performance even though the labelling from the reward model is not as meaningful as the actions from the IDM.

Based on these insights, we propose to introduce a surrogate reward providing guidance signal for the agent on the replay buffer dataset. Due to the instability of adversarial training (Goodfellow et al.; Arjovsky et al.; Ho & Ermon)

and our focus on the pretraining paradigm, we opt to adopt the VIPER algorithm (Escontrela et al.). Instead of training a discriminator, VIPER trains a video prediction model on the demonstration datasets and treats the likelihood of the video prediction model as the reward for policy learning, i.e. $r_t^{\text{VIPER}} = \log p_\nu(o_t|o_{<t})$. Using this reward, we train the policy with a dreamer-style actor-critic algorithm based on imagination in the latent space of the world model (Hafner et al., a). In order to do this, we first need to modify the reconstruction term in Equation (1) by adding an extra term for decoding the VIPER reward, i.e. $\log p_\theta(r_t^{\text{VIPER}}|s_t)$. Then, we further train a value estimator $V_\xi(s_t)$ using TD(λ)-return estimates, i.e.

$$V_\xi^\lambda(s_t) = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}V_\xi^{(n)}(s_t) \tag{7}$$
$$\text{with } V_\xi^{(n)}(s_t) = \sum_{t'=t+1}^{t+n}\gamma^{t'-t-1}r_{t'}^{\text{VIPER}} + \gamma^n V_\xi(s_{t+n}).$$

Using this estimate, we optimise our value function by minimising the MSE, i.e.

$$\xi^* = \operatorname*{argmin}_\xi (V_\xi(s_t) - V_{\xi'}^\lambda(s_t))^2. \tag{8}$$

As is common practice, we use a target value network with parameters ξ' to stabilise training, whose parameters are updated using Polyak averaging with a learning rate τ in every iteration.

Using this value estimate, we extend the policy objective of Equation (5) to

$$\psi^* = \operatorname*{argmax}_\psi \mathbb{E}_{o\sim D_{\text{demo}}, s\sim q_{\phi,\theta}, a\sim\pi_\psi}[\text{ELBO}] + \beta\mathbb{E}_{\{o,a\}\sim D_{\text{online}}, s\sim q_\phi, a'\sim\pi_\psi, s'\sim p_\theta}[V_{\xi'}^\lambda(s')], \tag{9}$$

where β is a hyper-parameter for balancing the two terms. We set $\beta = 0.1$ by default based on the difference of default learning rate in AIME and Dreamer.

4. Experiments

We aim to answer the following questions: a) How does the proposed AIME-NoB compare with state-of-the-art methods on common benchmarks? b) How well does the proposed modification resolve the EKB and the DKB? c) How do different choices of hyper-parameters influence the results? In order to answer these questions, we design our experiments on DMC and Meta-World benchmarks.

4.1. Datasets

For the DMC benchmark, we choose nine tasks across six embodiments following Liu et al. (a) and use their published dataset as the demonstration datasets. Each dataset contains only 10 trajectories to reflect the scarcity of expert demonstrations. For the embodiment dataset, in order not to leak the task information from the pretraining phase, we follow Rajeswar et al. and run a Plan2Explore (Sekar et al.) agent for each embodiment with 2M environments steps and use its replay buffer as the embodiment dataset. Different to them taking the model directly from the Plan2Explore agent as the pretrained model, we follow Zhang et al. to retrain the model for 200k gradient steps to get a better model.

For Meta-World benchmark, we use the data and model from Hansen et al. (a). The embodiment dataset was created from the replay buffer datasets. The open-sourced replay buffer datasets contain 40k trajectories for each of the 50 tasks with only state information. In order to fit to our image observation setup, we render the images by resetting the environment to the initial state of each trajectory and then executing the action sequence. The details can be found in Appendix E.

Following the idea of not leaking too much about the task information, inspired by the common practice in offline RL benchmarks (Fu et al.), we use the first 200 trajectories from each replay buffer and form a dataset with 10k trajectories in total. We call this dataset *MW-mt50*. To further study the out-of-distribution transfer ability of the pretrained model, we follow the difficulty classification of the tasks from (Seo et al., a) and only use the 39 easy and medium difficulty tasks to generate the datasets and the 11 tasks hard and very hard tasks as hold-out tasks. We uniformly sample 250 trajectories from the first 10k trajectories from each of the 39 tasks and form a dataset with 9750 trajectories in total. We refer to this dataset as *MW-mt39*. Hence, *MW-mt39* contains some expert behaviour solving the tasks, while *MW-mt50* consists of mostly exploratory behaviour.

As the demonstration datasets, we use the single-task policies open-sourced by TD-MPC2 and collect 50 trajectories for each tasks. We ensure that every trajectory in the demonstration dataset is successful. Since there are 500 steps in a DMC trajectory and only 100 steps in a Meta-World trajectory, the resulting datasets are roughly the same size.

4.2. Implementation

For the world model, we use the RSSM architecture (Hafner et al., b) with the hyper-parameters in Hafner et al. (a) for DMC tasks. In addition, we use the KL Balancing trick from Hafner et al. (c) to make the training more stable. For Meta-World, since the visual scene is more complex, we use the M size model from Hafner et al. (d), but still with the continuous latent variable to be aligned with other models used in this paper. The policy network is implemented with a two-layer MLP, with 128 neurons for each hidden layer. All the models are trained with Adam optimiser (Kingma &

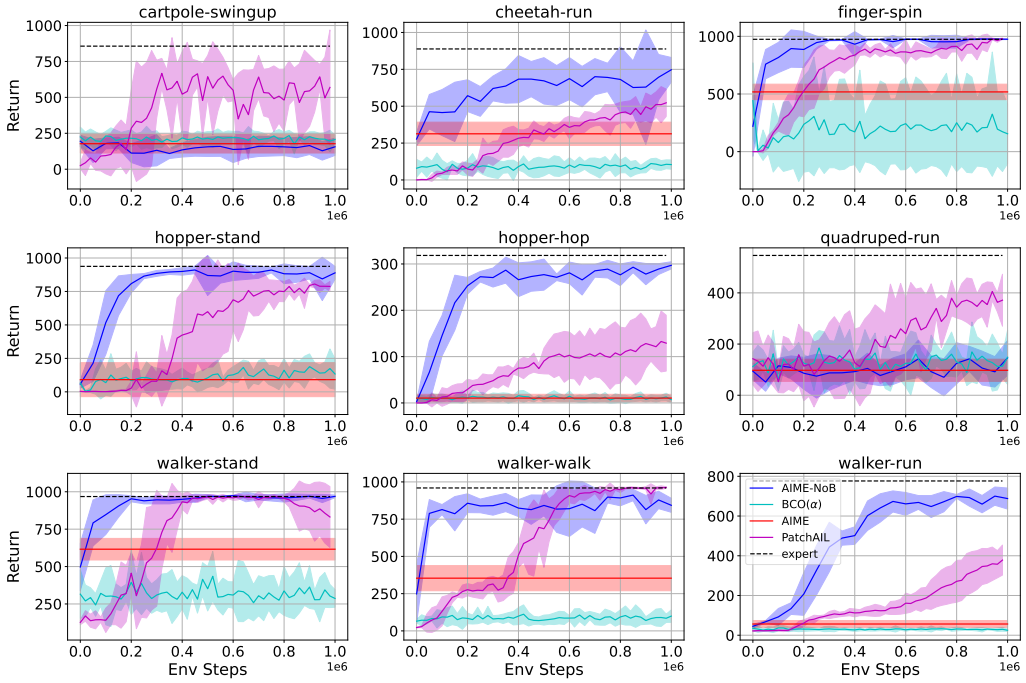


Figure 2. Benchmark results and 9 DMC tasks. Return are calculated by running the policy 10 times with the environment and taking the average return. The results are averaged across 5 seeds with the shade region representing 95% CI.

Ba). More details about the hyper-parameters can be found in Appendix C.

For the VIPER model, in the original paper, the authors first pretrain a VQ-GAN (Esser et al.) and then train a GPT-style auto-regressive model in the quantised space for prediction. For simplicity of the implementation, in this paper, we consider training an unconditioned latent world model as in Seo et al. (b) to model the VIPER reward. We use the same RSSM architecture of the model learning for DMC, only removing the condition of the actions, and we train the VIPER model for each task separately. Especially during training, we find training such a powerful model from scratch on a small dataset can easily result in over-fitting. Thus, we empirically choose to train the model only for 500 gradient steps for DMC models and 1000 gradient steps for Meta-World models. We show evidence of overfitting in Appendix F. Due to the large scale of the ELBO, we also apply symlog (Hafner et al., d) when computing the VIPER reward. Another difference with the original VIPER paper is that we do not use intrinsic motivation as the exploration bonus as the authors suggested, since the AIME loss for policy learning already provides task-related guidance for exploration. We only apply an entropy regulariser to the policy as is common practice. We further show the synergy between AIME and VIPER in Appendix G.

4.3. Benchmark Results

The benchmark results of DMC are shown in Figure 2. We compare AIME-NoB with AIME (Zhang et al.), BCO(α) (Torabi et al., a) and PatchAIL (Liu et al., a), a GAIL style algorithm. AIME-NoB significantly outperforms the PatchAIL baseline in 7 out of 9 tasks in terms of sample efficiency. Benefiting from the pretrained world model, AIME-NoB typically can reach expert performance within 200k environment steps. Compared with BCO(α), updating the model is regularised and is benefiting more from the online interaction to resolve the EKB. Compared with the original offline AIME, AIME-NoB reliably improves performance, especially in hard tasks such as walker-run and hopper where offline AIME did not manage to make any progress.

However, there are still two tasks for which AIME-NoB does not show much progress, namely cartpole-swingup and quadruped-run. For cartpole-swingup, we observe that the policy learns to move the cart out of the scene so that the static image yields a high likelihood from the video prediction model. A similar phenomenon was also discussed in the original VIPER paper (Escontrela et al.). For quadruped-run, we conjecture that it is due to visual difficulties of a reconstruction-based model. When the quadruped is initialised on the ground, due to the symmetric structure of the

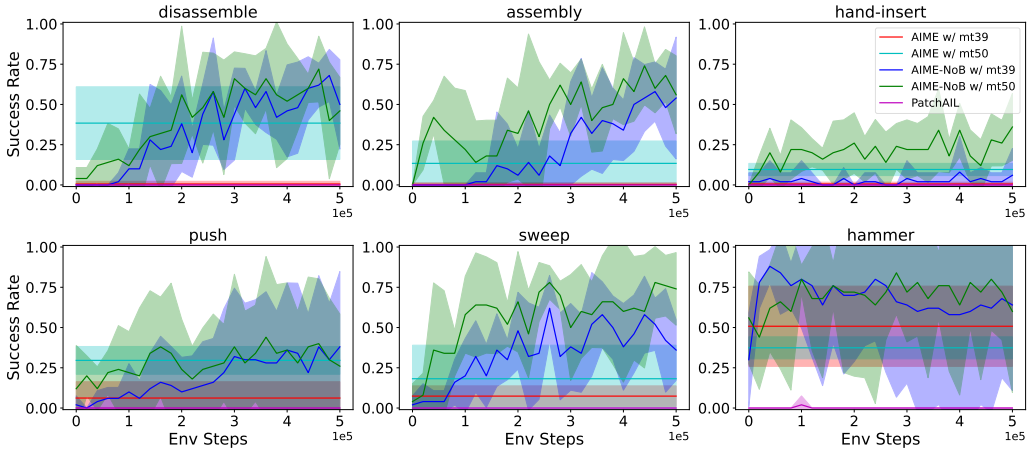


Figure 3. Benchmark results and 6 Meta-World tasks. Trajectories are only counted as success when it success at the last time steps and the success rates are calculated with 10 policy rollouts. The results are averaged across 5 seeds with the shade region representing 95% CI.

robot, it is impossible to figure out which action corresponds to which leg, and it easily leads the action inference process to diverge. We additionally show AIME-NoB can work on these tasks with the help of the true reward in Appendix G.

The benchmark results of Meta-World are shown in Figure 3. We choose four hard or very hard tasks, namely disassemble, assembly, hand-insert and push; and two medium difficult tasks, namely sweep and hammer. While PatchAIL does not work on these tasks at all, AIME and AIME-NoB can make progress on them. AIME-NoB using either of the pretrained models outperforms AIME in all tasks. On hard and very hard tasks, AIME with the mt50 model is better than AIME with the mt39 model. This is because they contain an unseen novel object by the mt39 model creating a large EKB. But in the online setting of AIME-NoB, the two models are mostly on par. Moreover, using the mt50 models is better than using the mt39 models on average, which may imply covering diverse behaviour is more valuable than knowing the expert directly.

4.4. Ablation Results

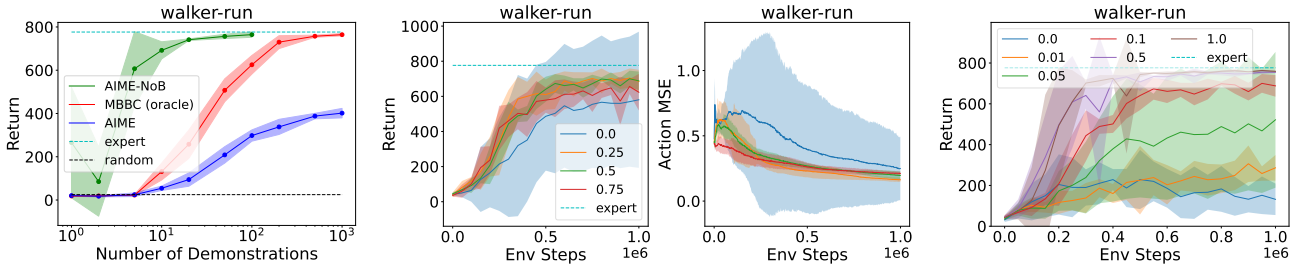
We conduct our ablation studies on walker-run task from DMC.

How well does the proposed methods resolve knowledge barriers? In order to show how well AIME-NoB resolves the two knowledge barriers, we the same experiment as in Figure 1 by providing the agent with different numbers of demonstrations. The result is shown in Figure 4a. As we discussed before, MBBC as an oracle method that circumvents the EKB is a strict upper bound for AIME. And AIME-NoB which addresses both the EKB and DKB achieves much better results and is an upper bound for MBBC. From AIME-NoB can achieve near-expert performance with as few as 5 demonstrations for this challenging task.

Influence of the data regulariser ratio α. We set the regulariser ratio α from [0.0, 0.25, 0.5, 0.75] and plot the results in Figure 4b. As we can see from the result, as long as we enable the regulariser, i.e. set $\alpha > 0$, we get reliable improvements over the course of training. But if we disable the regulariser by setting $\alpha = 0$, the learning exhibits high variance. In some cases, it fails to work entirely, while in others, learning only begins once sufficient new data accumulates in the replay buffer. As we discussed in Section 3.1, without the regulariser, in the early stage of the training, the model can easily overfit to the replay buffer, and it explains the early flattening phase of the training. As the training progresses, more and more data is available from the replay buffer, and it can establish the regulariser on its own, which explains the dramatic growing phase of the curves.

We also plot the MSE between the inferred actions and the true actions during the training process. From that we can see that a higher regulariser ratio offers more stable inference of the actions in the early phase of training.

Influence of the value gradient loss weight β. We set the weight β from [0.0, 0.01, 0.05, 0.1, 0.5, 1.0] and plot the results in Figure 4c. As the result shows, having a small β slows learning progress toward convergence. On the other hand, setting β to a much larger value will improve the sample-efficiency without causing instability. For the sample efficiency, since we only have 10 demonstrations, DKB dominates over EKB as shown in Figure 4a. Thus, having a larger β will make the learning much faster. In terms of the stability, as we discussed in 3.2, AIME loss and the value gradient loss operate on different regions of the



(a) Demonstration efficiency. (b) Ablation of α. (c) Ablation of β.

Figure 4. (a) Performance of AIME-NoB, MBBC, AIME w.r.t. different number of demonstrations. For AIME-NoB, we do not show the result for more than 100 demonstrations since it is already saturated to the expert. (b) Ablation for the choice of the regulariser ratio α. The left figure shows the mean return over 10 trajectories while the right figure shows the MSE between the inferred actions and the true actions. (c) Ablation for the choice of the weight of the value gradient loss β. All results are averaged across 5 seeds with the shaded region representing a 95% CI.

environment states. This could make their influence on the policy independent of each other.

5. Related Work

Imitation Learning from Observation. ILfO (Torabi et al., a;b; DeMoss et al.; Li et al.; Baker et al.; Zhang et al.; Liu et al., a) has become more popular in recent years due to their potential to utilise internet-scale videos for behaviour learning. Most of the previous works (Torabi et al., a;b; Li et al.) study the problem only with the true state as observation. Recent works (DeMoss et al.; Baker et al.; Zhang et al.; Liu et al., a) have started to shift toward image observations as a more general setting. Our work is a continuation of this journey.

Pretrained Models for Decision-Making. Inspired by the tremendous progress made in recent years in CV and NLP fields with the power of pretrained models, the decision-making community is also trying to follow the trend. Most recent works focus on the use of Large Language Model (LLM) for decision-making. A prompted model is used for producing trajectories and plans (Chen et al., 2023a; Huang et al., 2022; Ahn et al.; Di Palo et al., 2023), code (Vemprala et al., 2023; Liang et al.; Singh et al., 2022; Chen et al., 2023b; Huang et al., 2023) or for modifying the reward (Ma et al., 2023; Mahmoudieh et al., 2022). There are also other people studying the benefit of pretrained visual models for visuomotor tasks (Shah & Kumar; Majumdar et al.; Hansen et al., b; Parisi et al.) while others try to train large policy networks directly with transformers (Vaswani et al.) and huge datasets (Brohan et al., b;a; Reed et al.). However, there is only little attention being put on pretrained world models (Zhang et al.; Rajeswar et al.; Sekar et al.), which are natively developed by the model-based decision-making community and perfectly fit into the pretraining and finetuning paradigm. Our work explores this overlooked domain and showcases its potential.

6. Discussion

In this paper, we identify two knowledge barriers, namely the EKB and the DKB, which as we show limit the performance of state-of-the-art ILfO methods using pretrained models. We thoroughly analyse the underlying cause of each barrier and propose practical solutions. Specifically, we propose to use online interaction with a data-driven regulariser to solve the EKB and surrogate reward labelling to reduce the DKB. Combining these solutions, we propose AIME-NoB and showcase its efficiency compared to state-of-the-art ILfO methods. Our ablation studies show how each knowledge barrier is addressed by the proposed solution and how their hyper-parameters influence the performance.

However, the proposed solutions still have drawbacks. First, the data-driven regulariser is not practical when the model is pretrained on huge datasets – cf. foundation models popular in the fields of CV and NLP. Reducing the amount of data needed for the regulariser could greatly improve the usability of the method. Second, although having pretrained models is beneficial, having too many pretrained components can be detrimental for model selection. Especially in AIME-NoB, the world model and the VIPER model share a very similar interface. Designing a shared model that can serve both interfaces could ease the use of the method. Last but not least, due to the high demand of computing resources, we only study the pretrained world model on a very small scale. It will be an interesting direction to study these model at larger scales.

We hope our work can shed some light on the future development of ILfO method and bring more attention to the great potential of pretrained world models.

Impact Statements

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Ho, D., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jang, E., Ruano, R. J., Jeffrey, K., Jesmonth, S., Joshi, N. J., Julian, R., Kalashnikov, D., Kuang, Y., Lee, K.-H., Levine, S., Lu, Y., Luu, L., Parada, C., Pastor, P., Quiambao, J., Rao, K., Rettinghouse, J., Reyes, D., Sermanet, P., Sievers, N., Tan, C., Toshev, A., Vanhoucke, V., Xia, F., Xiao, T., Xu, P., Xu, S., Yan, M., and Zeng, A. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. URL http://arxiv.org/abs/2204.01691.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein Generative Adversarial Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 214–223. PMLR. URL https://proceedings.mlr.press/v70/arjovsky17a.html.

Baker, B., Akkaya, I., Zhokhov, P., Huizinga, J., Tang, J., Ecoffet, A., Houghton, B., Sampedro, R., and Clune, J. Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos. URL http://arxiv.org/abs/2206.11795.

Ball, P. J., Smith, L., Kostrikov, I., and Levine, S. Efficient online reinforcement learning with offline data. *arXiv preprint arXiv:2302.02948*, 2023.

Becker-Ehmck, P., Peters, J., and van der Smagt, P. Switching Linear Dynamics for Variational Bayes Filtering. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 553–562. PMLR. URL https://proceedings.mlr.press/v97/becker-ehmck19a.html.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X. L., Li, X., Ma, T., Malik, A., Manning, C. D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J. C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J. S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J., Ré, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K., Tamkin, A., Taori, R., Thomas, A. W., Tramèr, F., Wang, R. E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S. M., Yasunaga, M., You, J., Zaharia, M., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K., and Liang, P. On the Opportunities and Risks of Foundation Models. URL http://arxiv.org/abs/2108.07258.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., Florence, P., Fu, C., Arenas, M. G., Gopalakrishnan, K., Han, K., Hausman, K., Herzog, A., Hsu, J., Ichter, B., Irpan, A., Joshi, N., Julian, R., Kalashnikov, D., Kuang, Y., Leal, I., Lee, L., Lee, T.-W. E., Levine, S., Lu, Y., Michalewski, H., Mordatch, I., Pertsch, K., Rao, K., Reymann, K., Ryoo, M., Salazar, G., Sanketi, P., Sermanet, P., Singh, J., Singh, A., Soricut, R., Tran, H., Vanhoucke, V., Vuong, Q., Wahid, A., Welker, S., Wohlhart, P., Wu, J., Xia, F., Xiao, T., Xu, P., Xu, S., Yu, T., and Zitkovich, B. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. a.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jackson, T., Jesmonth, S., Joshi, N. J., Julian, R., Kalashnikov, D., Kuang, Y., Leal, I., Lee, K.-H., Levine, S., Lu, Y., Malla, U., Manjunath, D., Mordatch, I., Nachum, O., Parada, C., Peralta, J., Perez, E., Pertsch, K., Quiambao, J., Rao, K., Ryoo, M., Salazar, G., Sanketi, P., Sayed, K., Singh, J., Sontakke, S., Stone, A., Tan, C., Tran, H., Vanhoucke, V., Vega, S., Vuong, Q., Xia, F., Xiao, T., Xu, P., Xu, S., Yu, T., and Zitkovich, B. RT-1: Robotics Transformer for Real-World Control at Scale. b.

Chen, Y., Arkin, J., Dawson, C., Zhang, Y., Roy, N., and Fan, C. AutoTAMP: Autoregressive Task and Motion Planning with LLMs as Translators and Checkers, September 2023a. URL http://arxiv.org/abs/2306.06531. arXiv:2306.06531 [cs].

Chen, Y., Gandhi, R., Zhang, Y., and Fan, C. NL2TL: Transforming Natural Languages to Temporal Logics using Large Language Models, May 2023b. URL http://arxiv.org/abs/2305.07766. arXiv:2305.07766 [cs].

DeMoss, B., Duckworth, P., Hawes, N., and Posner, I. DITTO: Offline Imitation Learning with World Models. URL http://arxiv.org/abs/2302.03086.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. URL http://arxiv.org/abs/1810.04805.

Di Palo, N., Byravan, A., Hasenclever, L., Wulfmeier, M., Heess, N., and Riedmiller, M. Towards A Unified Agent with Foundation Models, July 2023. URL http://arxiv.org/abs/2307.09668. arXiv:2307.09668 [cs].

Escontrela, A., Adeniji, A., Yan, W., Jain, A., Peng, X. B., Goldberg, K., Lee, Y., Hafner, D., and Abbeel, P. Video Prediction Models as Rewards for Reinforcement Learning. URL https://openreview.net/forum?id=HWNl9PAYIP.

Esser, P., Rombach, R., and Ommer, B. Taming Transformers for High-Resolution Image Synthesis. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12868–12878. IEEE. ISBN 978-1-66544-509-2. doi: 10.1109/CVPR46437.2021.01268. URL https://ieeexplore.ieee.org/document/9578911/.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4RL: Datasets for Deep Data-Driven Reinforcement Learning. URL http://arxiv.org/abs/2004.07219.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper_files/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html.

Ha, D. and Schmidhuber, J. Recurrent World Models Facilitate Policy Evolution. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2018/file/2de5d16682c3c35007e4e92982f1a2ba-Paper.pdf.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to Control: Learning Behaviors by Latent Imagination. In *ICLR 2020*, a. URL https://openreview.net/forum?id=S1lOTC4tDS.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2555–2565. PMLR, b. URL https://proceedings.mlr.press/v97/hafner19a.html.

Hafner, D., Lillicrap, T. P., Norouzi, M., and Ba, J. Mastering Atari with Discrete World Models. c. URL https://openreview.net/forum?id=0oabwyZbOu.

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. Mastering Diverse Domains through World Models, d. URL http://arxiv.org/abs/2301.04104.

Hansen, N., Su, H., and Wang, X. TD-MPC2: Scalable, Robust World Models for Continuous Control. In *ICLR 2024*, a. URL https://openreview.net/forum?id=Oxh5CstDJU.

Hansen, N., Yuan, Z., Ze, Y., Mu, T., Rajeswaran, A., Su, H., Xu, H., and Wang, X. On Pre-Training for Visuo-Motor Control: Revisiting a Learning-from-Scratch Baseline, b. URL http://arxiv.org/abs/2212.05749.

He, K., Chen, X., Xie, S., Li, Y., Dollar, P., and Girshick, R. Masked Autoencoders Are Scalable Vision Learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15979–15988. IEEE, a. ISBN 978-1-66546-946-3. doi: 10.1109/CVPR52688.2022.01553. URL https://ieeexplore.ieee.org/document/9879206/.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. IEEE, b. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.90. URL http://ieeexplore.ieee.org/document/7780459/.

Ho, J. and Ermon, S. Generative Adversarial Imitation Learning. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper_files/paper/2016/hash/cc7e2b878868cbae992d1fb743995d8f-Abstract.html.

Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., Sermanet, P., Brown, N., Jackson, T., Luu, L., Levine, S., Hausman, K., and Ichter, B. Inner Monologue: Embodied Reasoning through Planning with Language Models, July 2022. URL http://arxiv.org/abs/2207.05608. arXiv:2207.05608 [cs].

Huang, W., Wang, C., Zhang, R., Li, Y., Wu, J., and Fei-Fei, L. VoxPoser: Composable 3D Value Maps for Robotic Manipulation with Language Models, November 2023. URL http://arxiv.org/abs/2307.05973. arXiv:2307.05973 [cs].

Karl, M., Soelch, M., Bayer, J., and van der Smagt, P. Deep Variational Bayes Filters: Unsupervised Learning of State Space Models from Raw Data. URL http://arxiv.org/abs/1605.06432.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In *3rd International Conference for Learning Representations*. doi: 10.48550/arXiv.1412.6980. URL http://arxiv.org/abs/1412.6980.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. URL http://arxiv.org/abs/1312.6114.

Klushyn, A., Kurle, R., Soelch, M., Cseke, B., and van der Smagt, P. Latent Matters: Learning Deep State-Space Models. URL https://openreview.net/forum?id=-WEryOMRpZU#all.

Lee, S., Seo, Y., Lee, K., Abbeel, P., and Shin, J. Offline-to-Online Reinforcement Learning via Balanced Replay and Pessimistic Q-Ensemble. In *Proceedings of the 5th Conference on Robot Learning*, pp. 1702–1712. PMLR. URL https://proceedings.mlr.press/v164/lee22d.html.

Li, A., Boots, B., and Cheng, C.-A. MAHALO: Unifying Offline Reinforcement Learning and Imitation Learning from Observations. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org. URL http://arxiv.org/abs/2303.17156.

Liang, J., Huang, W., Xia, F., Xu, P., Hausman, K., Ichter, B., Florence, P., and Zeng, A. Code as Policies: Language Model Programs for Embodied Control. URL http://arxiv.org/abs/2209.07753.

Liu, M., He, T., Zhang, W., Yan, S., and Xu, Z. Visual Imitation Learning with Patch Rewards. a. URL https://openreview.net/forum?id=OnM3R47KIiU.

Liu, Y.-R., Hu, Y.-Q., Qian, H., Qian, C., and Yu, Y. ZOOpt: Toolbox for Derivative-Free Optimization. 65 (10):207101, s11432–021–3416–y, b. ISSN 1674-733X, 1869-1919. doi: 10.1007/s11432-021-3416-y. URL http://arxiv.org/abs/1801.00329.

Ma, Y. J., Liang, W., Wang, G., Huang, D.-A., Bastani, O., Jayaraman, D., Zhu, Y., Fan, L., and Anandkumar, A. Eureka: Human-Level Reward Design via Coding Large Language Models, October 2023. URL http://arxiv.org/abs/2310.12931. arXiv:2310.12931 [cs].

Mahmoudieh, P., Pathak, D., and Darrell, T. Zero-Shot Reward Specification via Grounded Natural Language. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 14743–14752. PMLR, June 2022. URL https://proceedings.mlr.press/v162/mahmoudieh22a.html. ISSN: 2640-3498.

Majumdar, A., Yadav, K., Arnaud, S., Ma, Y. J., Chen, C., Silwal, S., Jain, A., Berges, V.-P., Abbeel, P., Batra, D., Lin, Y., Maksymets, O., Rajeswaran, A., and Meier, F. Where are we in the search for an Artificial Visual Cortex for Embodied Intelligence? URL https://openreview.net/forum?id=NJtSbIWmt2T.

Nakamoto, M., Zhai, Y., Singh, A., Mark, M. S., Ma, Y., Finn, C., Kumar, A., and Levine, S. Cal-QL: Calibrated Offline RL Pre-Training for Efficient Online Fine-Tuning. URL https://openreview.net/forum?id=GcEIvidYSw.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback. URL http://arxiv.org/abs/2203.02155.

Parisi, S., Rajeswaran, A., Purushwalkam, S., and Gupta, A. The Unsurprising Effectiveness of Pre-Trained Vision Models for Control. URL http://arxiv.org/abs/2203.03580.

Peng, X. B., Ma, Z., Abbeel, P., Levine, S., and Kanazawa, A. AMP: Adversarial motion priors for stylized physics-based character control. 40(4):1–20. ISSN 0730-0301, 1557-7368. doi: 10.1145/3450626.3459670. URL https://dl.acm.org/doi/10.1145/3450626.3459670.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8748–8763. PMLR, a. URL https://proceedings.mlr.press/v139/radford21a.html.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language Models are Unsupervised Multitask Learners. pp. 24, b.

Rajeswar, S., Mazzaglia, P., Verbelen, T., Piché, A., Dhoedt, B., Courville, A., and Lacoste, A. Mastering the Unsupervised Reinforcement Learning Benchmark from Pixels. URL http://arxiv.org/abs/2209.12016.

Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J. T., Eccles, T., Bruce, J., Razavi, A., Edwards, A., Heess, N., Chen, Y., Hadsell, R., Vinyals, O., Bordbar, M., and de Freitas, N. A Generalist Agent. URL http://arxiv.org/abs/2205.06175.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. URL http://arxiv.org/abs/1401.4082.

Ross, S., Gordon, G., and Bagnell, D. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings. URL https://proceedings.mlr.press/v15/ross11a.html.

Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., and Pathak, D. Planning to Explore via Self-Supervised World Models. URL http://arxiv.org/abs/2005.05960.

Seo, Y., Hafner, D., Liu, H., Liu, F., James, S., Lee, K., and Abbeel, P. Masked World Models for Visual Control, a. URL http://arxiv.org/abs/2206.14244.

Seo, Y., Lee, K., James, S., and Abbeel, P. Reinforcement Learning with Action-Free Pre-Training from Videos. In *ICML*, volume 162, pp. 19561–19579. PMLR, b. doi: 10.48550/arXiv.2203.13880.

Shah, R. and Kumar, V. RRL: Resnet as representation for Reinforcement Learning. URL http://arxiv.org/abs/2107.03380.

Singh, I., Blukis, V., Mousavian, A., Goyal, A., Xu, D., Tremblay, J., Fox, D., Thomason, J., and Garg, A. ProgPrompt: Generating Situated Robot Task Plans using Large Language Models, September 2022. URL http://arxiv.org/abs/2209.11302. arXiv:2209.11302 [cs].

Torabi, F., Warnell, G., and Stone, P. Behavioral Cloning from Observation, a. URL http://arxiv.org/abs/1805.01954.

Torabi, F., Warnell, G., and Stone, P. Generative Adversarial Imitation from Observation, b. URL http://arxiv.org/abs/1807.06158.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. LLaMA: Open and Efficient Foundation Language Models, a. URL http://arxiv.org/abs/2302.13971.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open Foundation and Fine-Tuned Chat Models, b. URL http://arxiv.org/abs/2307.09288.

Tunyasuvunakool, S., Muldal, A., Doron, Y., Liu, S., Bohez, S., Merel, J., Erez, T., Lillicrap, T., Heess, N., and Tassa, Y. Dm_control: Software and tasks for continuous control. 6:100022. ISSN 2665-9638. doi: 10.1016/j.simpa.2020.100022. URL https://www.sciencedirect.com/science/article/pii/S2665963820300099.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Vemprala, S., Bonatti, R., Bucker, A., and Kapoor, A. ChatGPT for Robotics: Design Principles and Model Abilities, July 2023. URL http://arxiv.org/abs/2306.17582. arXiv:2306.17582 [cs].

Yu, T., Quillen, D., He, Z., Julian, R., Narayan, A., Shively, H., Bellathur, A., Hausman, K., Finn, C., and Levine, S. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. URL http://arxiv.org/abs/1910.10897.

Zhang, X., Becker-Ehmck, P., van der Smagt, P., and Karl, M. Action Inference by Maximising Evidence:

Zero-Shot Imitation from Observation with World Models. URL https://openreview.net/forum?id=WjlCQxpuxU.

A. Algortihm

Algorithm 1 AIME-NoB

Input: Embodiment dataset D_{body}, Demonstration dataset D_{demo}, Pretrained world model parameters ϕ, θ, Pretrained VIPER model p_ν, regulariser ratio α, value gradient weight β, batch size B
Initialise policy and critic parameters ψ, ξ randomly.
for $i = 1$ **to** policy pretraining iterations **do**
  Draw a batch of demonstrations $o_{1:T} \sim D_{\text{demo}}$
  Update policy parameters ψ with Equation (5).
end for
Initialize $D_{\text{online}} \to \emptyset$.
for $i = 1$ **to** Environment Interaction budget **do**
  Collect a new episode $\{o_{1:T}, a_{1:T}\}$ with the current policy π_ψ
  Estimate reward using VIPER $r^{\text{VIPER}}_{1:T} = p_\nu(o_{1:T})$
  Append $\{o_{1:T}, a_{1:T}, r^{\text{VIPER}}_{1:T}\}$ to D_{online}
  # Update world model
  Draw $\alpha \cdot \text{B}$ samples $b_{\text{body}} \sim D_{\text{body}}$
  Draw $(1-\alpha) \cdot \text{B}$ samples $b_{\text{online}} \sim D_{\text{online}}$
  Define combined batch $b = b_{\text{body}} \cup b_{\text{online}}$
  Finetune model with batch b using Equation (6).
  # Update policy
  Sample a batch from D_{demo}
  Update policy parameters ψ with Equation (9).
  Update value function parameters ξ with Equation (8).
end for

B. Compute Resources

All the experiments are run on a local cluster with a few A100 and RTX8000 instances. All the experiments are tuned to use less than 10GB of GPU memory so that they can run in A100 MIG. World models pretraining requires about 24 GPU hours, while VIPER models require negligible time for training. Each DMC experiment requires about 40 GPU hours while each Meta-World experiment requires about 24 GPU hours.

C. Hyper-parameters

Here, we document the detailed hyper-parameters for all the trained models in Table 1.

D. Source of Datasets

We use the expert trajectories from Liu et al. (a) at `https://osf.io/4w69f/?view_only=e29b9dc9ea474d038d533c2245754f0c`. The authors didn't provide a License for their dataset. Besides, we use the replay buffer dataset from Hansen et al. (a) at `https://huggingface.co/datasets/nicklashansen/tdmpc2/tree/main/mt80`. The authors provide the dataset under the MIT License. Moreover, we use the replay buffer dataset from Zhang et al. at `https://github.com/argmax-ai/aime/tree/main/datasets`. The authors provide the dataset under the CC BY 4.0 License.

E. Details for Resetting Meta-World Tasks

To generate the image observation datasets from the TD-MPC2 replay buffer (Hansen et al., a), we modify the Meta-World codebase to reset the environment to the initial state of the trajectory from the first observation. Luckily, the starting position of the robot arm is always the same for each task, so that we do not need to apply inverse kinematics to solve for the initial pose of the robot arm. For the object and the target position, for most of the tasks, the internal reset position can be computed by making a constant shift on the object position and the target position in the observations. There are, however, also a few

edge cases which we handle differently.

In button-press-topdown and button-press-topdown-wall, the object's true position only appears in the observation upon the second time step, presumably due to some simulator delay in the resetting process. So for these two tasks, the initial state is reset by the second observation.

For basketball and box-close, it seems like there is some internal collision detection that will alter the object and robot position after the task is reset, so computing the exact reset value from the observation is not possible. For these two tasks, we instead resort to a search-based method. To be specific, we use a gradient-free optimiser from (Liu et al., b) to search over the resetting space of the object and find the reset position that minimises the L2 distance with the true observation.

More details of the implementation can be found in the code.

F. Overfitting of the VIPER Models

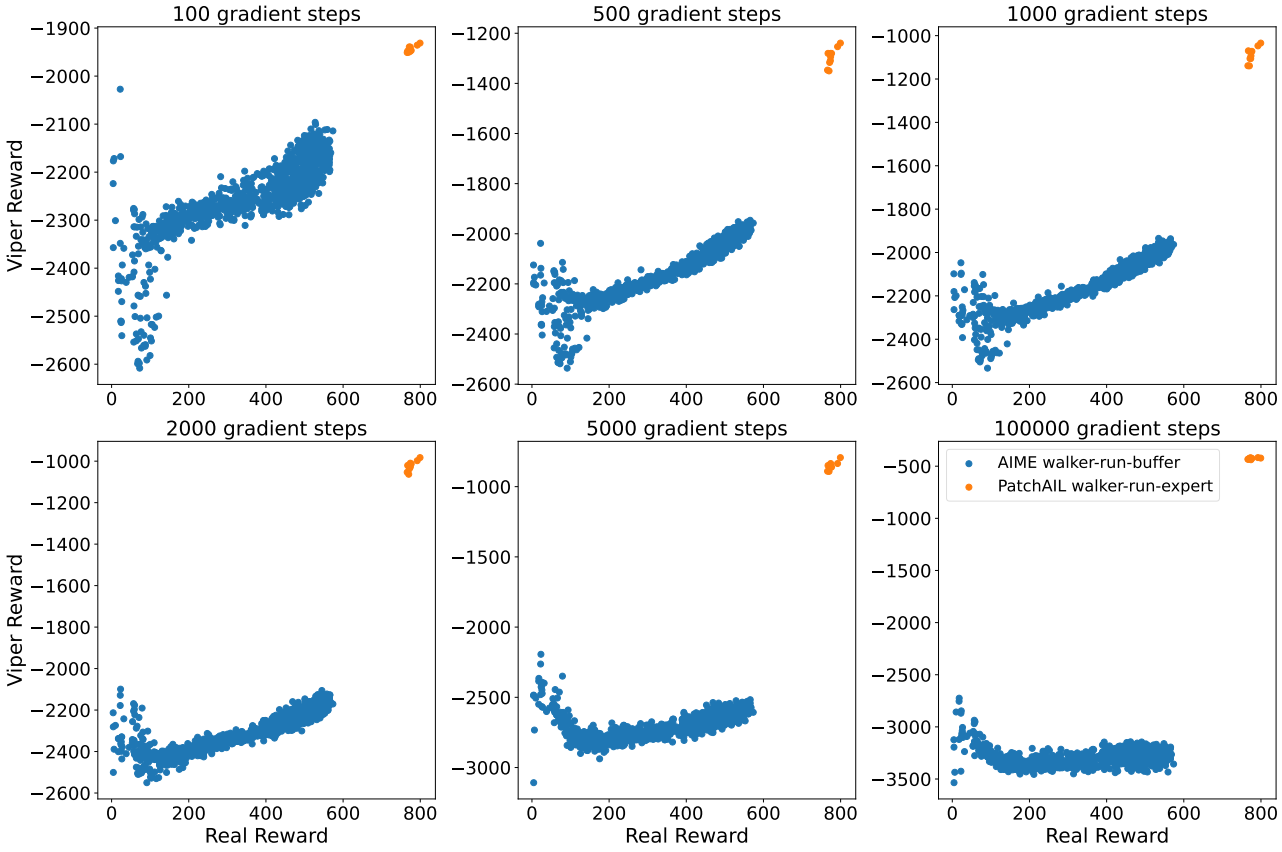


Figure 5. Correlation of the VIPER reward and the real reward with models trained with different numbers of gradient steps. Each point represents one trajectory. We can clearly see the model gradually overfitting and losing the correlation with the real reward when training for more than 1000 gradient steps.

To better illustrate the overfitting problem for VIPER models and justify our choice of training fewer iterations, we train the VIPER models for a varying number of gradient steps and evaluate the correlations between the VIPER reward and the true reward on both the expert dataset from PatchAIL, where the VIPER model is trained on, and the replay buffer dataset from Zhang et al..

Specifically, we train the same VIPER model with {100, 500, 1000, 2000, 5000, 100000} gradient steps and plot the result in Figure 5. As we can clearly see, when training with less than or equal to 1000 gradient steps, VIPER reward has a very nice correlation with the true reward, with the middle-range performance even like a linear correlation. The best model could

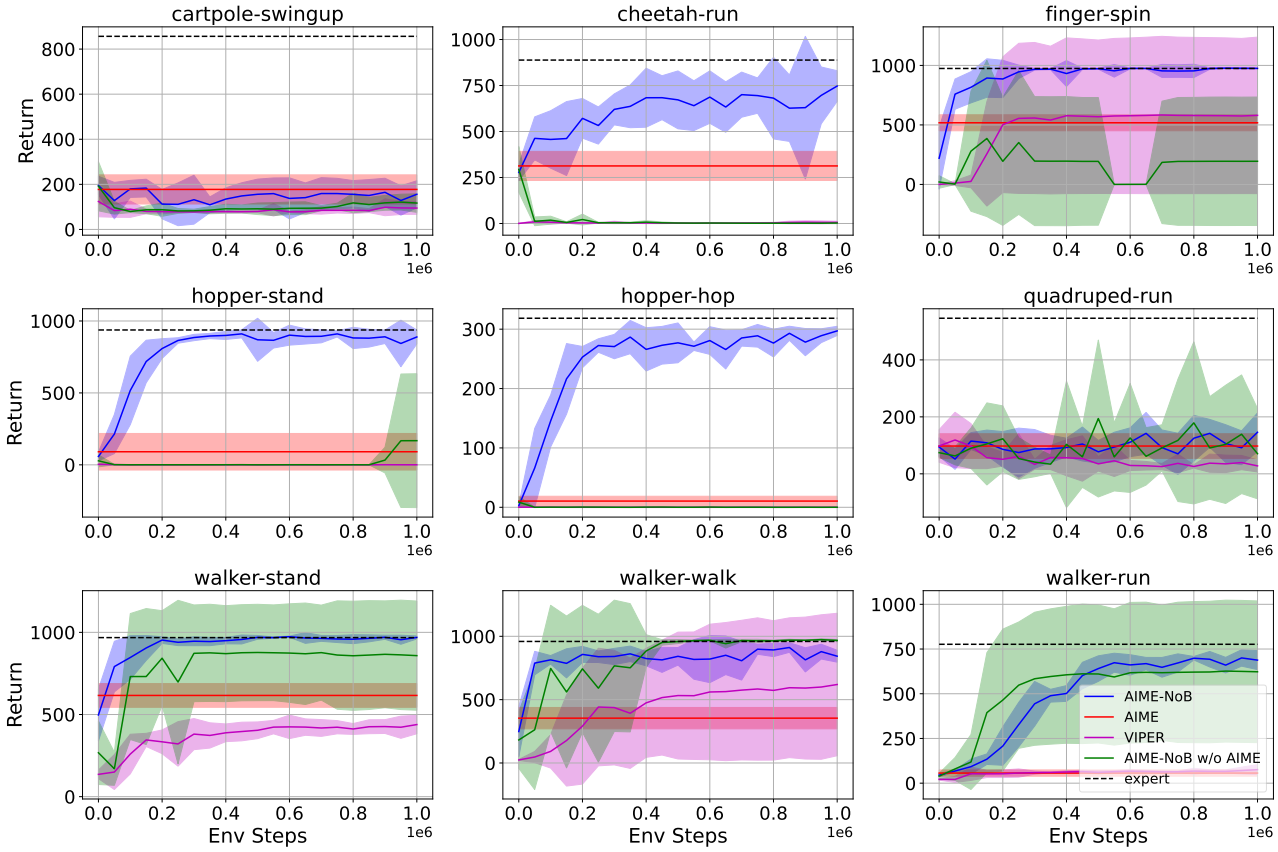


Figure 6. Additional ablation on DMC tasks by exploring the synergy between AIME and VIPER model. Return are calculated by running the policy 10 times with the environment and taking the average return. The results are averaged across 5 seeds with the shade region representing 95% CI.

be selected from 500 and 1000 gradient steps. However, as we train the model for longer, the VIPER reward for the expert trajectories is boosted even higher, and as a side effect, it also relatively boosts up the VIPER reward for low-performance trajectories. This is because, when overfitting the expert trajectories, the model increases the marginal likelihood of all the observations in the expert trajectories to a higher value, which also includes a few frames of the robot lying on the ground at the very beginning of each trajectory after reset. For these low-performance trajectories, the robot remains mainly stuck around the initial position and struggles on the ground. This artifact of the overfitted VIPER reward creates a sharp local maximum in the low-performance region that the agent can hardly get away from.

G. Additional Experiments

Synergy between AIME and VIPER model. We also find there is a synergy between AIME and VIPER model. As we showed in Appendix F, one inherent problem of VIPER reward is that it not only incentivises the expert behaviour as the optimal, but also a stationary behaviour with very low reward as a local maxima. In order to work with the VIPER reward, the agent needs to have the ability to escape from the local maxima region. AIME offers the IL loss to imitate the expert demonstrations and can achieve decent performance even when pretrained offline, which helps to escape the local maxima. To better show the synergy, we provide additional ablation results with the VIPER reward in Figure 6. In the experiments, we include two other variants: the AIME-NoB w/o AIME is to remove the AIME IL loss from the online policy learning, so that the policy is pretrained by AIME loss but finetuned with only RL loss from the VIPER reward; while the VIPER is following the implementation in the original VIPER paper with RL loss on both the VIPER reward for the task and intrinsic reward for exploration. From the result, we can clearly see that without the help of AIME IL loss, VIPER reward cannot

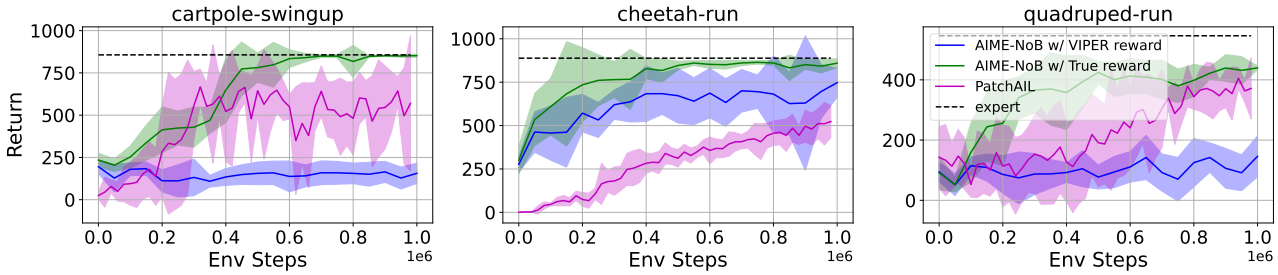


Figure 7. Additional benchmark results on 3 DMC tasks with an additional variant of AIME-NoB with the true reward. Return are calculated by running the policy 10 times with the environment and taking the average return. The results are averaged across 5 seeds with the shade region representing 95% CI.

reliably motivate the agent to learn good behaviours. Even when in walker tasks, the w/o AIME variant can solve the tasks to a certain extent; it depends strongly on the random seeds. In conclusion, the good performance of AIME-NoB cannot be achieved by either AIME IL loss or VIPER RL loss alone but by a combination of both.

Improving AIME-NoB with better rewards. We show additional results on the 3 not-so-well-performing DMC tasks, namely cartpole-swingup, cheetah-run and quadruped-run, in 7. In the plot we add a new variant using the true reward from the environment to replace the VIPER reward. As the results show, if we had a better estimation of the surrogate reward, AIME-NoB could also achieve good performance on these tasks.

Table 1. AIME-NoB hyper-parameters use for each benchmark.

	DMC	META-WORLD
WORLD MODEL		
CNN STRUCTURE	HA & SCHMIDHUBER	HAFNER ET AL. (D)
CNN WIDTH	32	48
MLP HIDDEN SIZE	512	640
MLP HIDDEN LAYER	2	3
MLP ACTIVATIONS	LAYERNORM + SWISH	
DETERMINISTIC LATENT SIZE	512	1024
STOCHASTIC LATENT SIZE	30	
FREE NATS	1.0	
KL BALANCING	0.8	
POLICY		
HIDDEN SIZE	128	
HIDDEN LAYER	2	
ACTIVATION	ELU	
DISTRIBUTION	TANH-GAUSSIAN	
VALUE NETWORK		
HIDDEN SIZE	128	
HIDDEN LAYER	2	
ACTIVATION	ELU	
TARGET EMA DECAY	0.95	
TRAINING		
BATCH SIZE	50	16
HORIZON	50	64
TOTAL ENV STEPS	1M	500k
UPDATE RATIO	0.1	
GRADIENT CLIP	100	
POLICY ENTROPY REGULARISER WEIGHT	1e-4	
MODEL LEARNING RATE	3e-4	
POLICY LEARNING RATE	3e-4	
VALUE NETWORK LEARNING RATE	8e-5	
DISCOUNT FACTOR γ	0.99	
TD-LAMBDA PARAMETER λ	0.95	
IMAGINE HORIZON	15	
AIME-NOB SPECIFIC		
POLICY PRETRAINING ITERATIONS	2000	
DATA-DRIVEN REGULARISER RATIO α	0.5	
VALUE GRADIENT LOSS WEIGHT β	0.1	