Collaborative World Models: An Online-Offline Transfer RL Approach

Qi Wang[1,2*]   **Junming Yang**[3*]   **Yunbo Wang**[1†]   **Xin Jin**[2]   **Wenjun Zeng**[2]   **Xiaokang Yang**[1]

[1] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai, China
[2] Eastern Institute for Advanced Study, Zhengjiang, China
[3] Nanjing University of Posts and Telecommunications, Nanjing, China

```
{qiwang067, yunbow, xkyang}@sjtu.edu.cn
              jmingyang@outlook.com
    {jinxin, wenjunzeng}@eias.ac.cn
```

Abstract

Training visual reinforcement learning (RL) models in offline datasets is challenging due to overfitting issues in representation learning and overestimation problems in value function. In this paper, we propose a transfer learning method called Collaborative World Models (CoWorld) to improve the performance of visual RL under offline conditions. The core idea is to use an easy-to-interact, off-the-shelf simulator to train an auxiliary RL model as the online "test bed" for the offline policy learned in the target domain, which provides a flexible constraint for the value function—Intuitively, we want to mitigate the overestimation problem of value functions outside the offline data distribution without impeding the exploration of actions with potential advantages. Specifically, CoWorld performs *domain-collaborative representation learning* to bridge the gap between online and offline hidden state distributions. Furthermore, it performs *domain-collaborative behavior learning* that enables the source RL agent to provide target-aware value estimation, allowing for effective offline policy regularization. Experiments show that CoWorld significantly outperforms existing methods in offline visual control tasks in DeepMind Control and Meta-World.

1 Introduction

Learning control policies in the physical world through visual observations can be challenging due to high environmental interaction costs and low sample efficiency [16, 28, 31, 35, 24, 18]. Offline reinforcement learning (RL) has emerged as a promising approach to overcome these challenges. However, applying offline RL in the visual world poses two main difficulties. First, offline visual RL, similar to the typical state-space offline RL setting [7, 15, 25, 2, 45], is prone to the overestimation problem in the value function (see the blue curve in Figure 1(a)) or underestimation (red curve) when overly constraining the values outside the offline distribution. Second, it may suffer from overfitting issues when extracting hidden states from the limited and partially observable visual data, which may result in trivial solutions of representation learning. In our preliminary experiments, we observed that most existing visual RL methods, such as CURL [16] and Dreamer [11, 13], underperformed in the offline setting. Improving offline visual RL remains a relatively under-explored area of research.

Our goal is to address the tradeoff between overestimation and over-conservatism in the value function of offline visual RL by using a *mildly-conservative* policy that does not impede the exploration of

---

*Equal contribution.
†Corresponding author: Yunbo Wang.

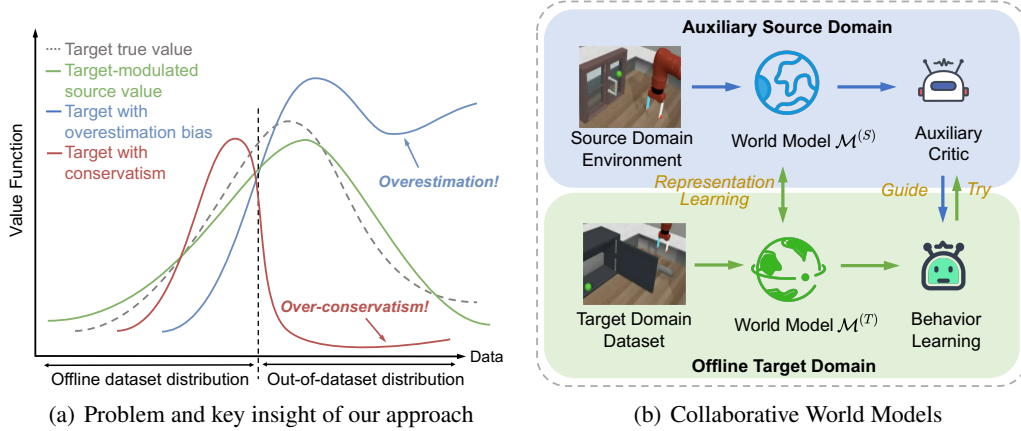|	
(a) Problem and key insight of our approach	(b) Collaborative World Models

Figure 1: CoWorld uses an auxiliary online environment to build a policy "test bed" that is aware of offline domain information. This, in turn, can guide the visual RL agent in the offline domain to learn a mildly-conservative policy, striking a balance between value overestimation and over-conservatism.

actions with potential advantages. To achieve this, as illustrated in Figure 1(b), we propose to *use an off-the-shelf, easy-to-interact visual control simulator to train an auxiliary model-based RL (MBRL) agent, which serves as the online "test bed" for the target policy learned in the offline domain*. This allows us to flexibly constrain the value function, thus mitigating the value overestimation problem while preserving the ability to explore actions with potential advantages. Another benefit of leveraging an online auxiliary domain is that the source world model can alleviate the overfitting issue of representation learning. This is because it greatly augments the observed visual data beyond the offline videos, producing more generalizable representations of the hidden states.

Specifically, we present a model-based transfer RL approach called **Co**llaborative **World** Models (CoWorld), which consists of two main training processes performed in an iterative manner: domain-collaborative representation learning and domain-collaborative behavior learning. During *domain-collaborative representation learning*, CoWorld bridges the gap between online and offline hidden state distributions through knowledge distillation. By leveraging the rich physical interactions with the online visual environment, it avoids trivial solutions in the inference of hidden states in the offline domain. For *domain-collaborative behavior learning*, CoWorld trains the source RL model to provide target-aware value estimation (green curve), allowing for effective offline policy regularization. To enable the source critic to evaluate the policy from the offline target domain, we train the reward predictor of the source model using mixed supervision from both the online environment and the offline dataset. The source critic acts as a regularizer for the learning process of the target value function in a mildly-conservative way, balancing between value overestimation and over-conservatism. In particular, we do not directly penalize the value function of state-action pairs that are out-of-dataset during behavior learning, but only when the values exceed the scope of the source value function.

In summary, this paper presents a novel approach to improve offline visual RL by leveraging an online auxiliary domain. The key technical contributions of this work include:

- We mitigate the overfitting issue in representation learning by training an auxiliary world model that can interact with the online source environment and close the gap between the hidden state distributions across domains. It also forms the basis for domain-collaborative behavior learning.
- We tackle the value overestimation issue by training an auxiliary critic in the online source domain to regularize value estimation in the offline target domain, using a flexible conservatism penalty.

Experiments demonstrate that CoWorld outperforms existing methods significantly in offline visual control tasks in the modified DeepMind Control and Meta-World environments.

2 Problem Formulation

We consider offline visual RL within the framework of partially observable Markov decision processes (POMDPs) and aim to use a source POMDP denoted by $\langle \mathcal{O}^{(S)}, \mathcal{A}, P^{(S)}, R^{(S)}, \gamma \rangle$ to facilitate policy
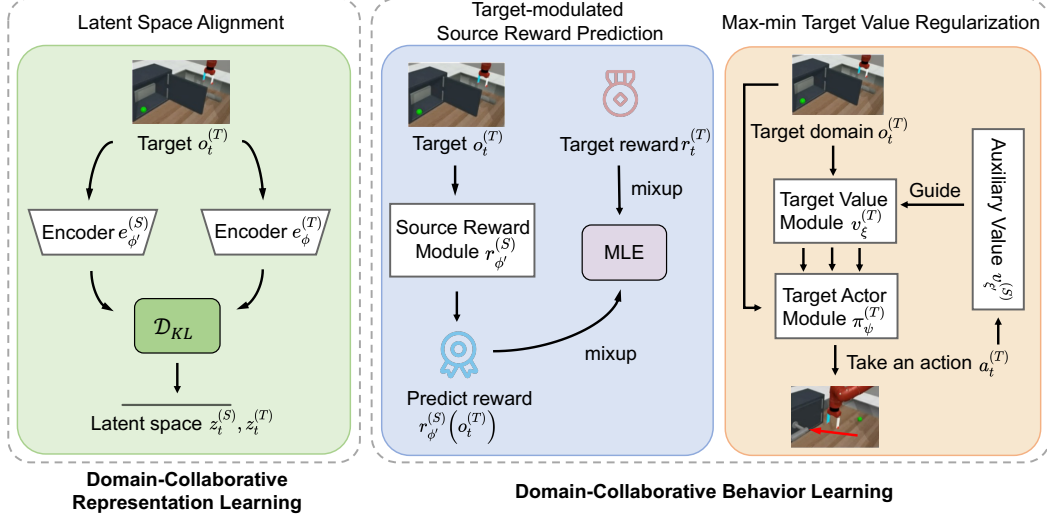
Figure 2: The overall transfer learning pipeline of CoWorld involves two model-based RL agents and three training stages. Please refer to the text in Section 3.1 for detailed descriptions.

learning in the target POMDP denoted by $\langle \mathcal{O}^{(T)}, \mathcal{A}, P^{(T)}, R^{(T)}, \gamma \rangle$. The observations and physical dynamics have different distributions across domains. We focus on the scenario where the source POMDP is available online, allowing the agent to interact with the environment and collect new data. In contrast, the target POMDP is offline, and the agent learns to maximize the expected discounted cumulative reward with a fixed dataset $\mathcal{B}^{(T)} = \{(o_t^{(T)}, a_t^{(T)}, r_t^{(T)}, o_{t+1}^{(T)})\}_{t=1}^{T}$.

3 Method

In Section 3.1, we present the general framework of CoWorld that solves offline visual RL as an online-offline transfer learning problem. In Section 3.2, we first discuss the pretraining procedure of the source model-based RL agent, and then propose the domain-collaborative representation learning method that closes the distance between the hidden state distributions of the source and target domains. We also highlight the advantages of this approach in tackling the aforementioned challenges in offline visual RL. In Section 3.3, we introduce the domain-collaborative behavior learning algorithm, which is the most crucial component in CoWorld. Concretely, we discuss two key aspects: enabling the source critic to evaluate the policy from the offline target domain, and using the source critic as a test bed to evaluate target domain policies and regularize its value function estimation.

3.1 Overview of CoWorld

To facilitate a more effective transfer of RL knowledge between online and offline domains, we have made improvements in two key aspects: (i) the world model learning with domain alignment and (ii) the online-offline behavior learning. As shown in Figure 2, the overall transfer learning pipeline of CoWorld involves two model-based RL agents and has three training stages. According to the overall framework shown in Alg. 1, we first initialize the source domain (S) and target (T) domain agent, including the world models $\{\mathcal{M}^{(S)}, \mathcal{M}^{(T)}\}$, the actor networks $\{\pi^{(S)}, \pi^{(T)}\}$ and the critic networks $\{v^{(S)}, v^{(T)}\}$. We initiate the training process by pretraining the agent in the source domain, followed by iterative training of both the target domain and source domain agents:

a) During domain-collaborative representation, we train $\mathcal{M}^{(T)}$ by aligning its latent-space distributions with those from $\mathcal{M}^{(S)}$, conditioned on the same target domain inputs (**Line 9**).

b) To enable the auxiliary agent to provide effective guidance for $\pi^{(T)}$ and $v^{(T)}$, we first transfer the prior information of the target domain into the source reward predictor (**Lines 19-21**) and then perform target-modulated behavior learning over the source domain imaginations (**Line 22**).

c) We train the offline visual RL agent by taking $v^{(S)}$ as the test bed for $\pi^{(T)}$, and use it to regularize $v^{(T)}$ in the offline target domain (**Lines 11-13**).

3

Algorithm 1: CoWorld.

---

1   **Hyperparameters:** H: Imagination horizon; K_1, K_2: Target domain and source domain update steps.
2   **Initialize:** $\phi, \phi', \psi, \psi', \xi, \xi'$: Source and target agent parameters; $\mathcal{B}^{(S)}, \mathcal{B}^{(T)}$: Replay buffers.
3   Pretrain the source domain agent and update its parameters ϕ', ψ', ξ'.
4   **while** *not converged* **do**
5   // Train the target domain agent
6   **for** *update step* $k_t = 1 \ldots K_1$ **do**
7   Draw a batch of trajectories from the offline buffer, $\{(o_t, a_t, r_t)\}_{t=1}^{T} \sim \mathcal{B}^{(T)}$.
8   // Domain-Collaborative Representation learning
9   Train the target world model $\mathcal{M}_{\phi}^{(T)}$ using Eq. (3) and update ϕ.
10   // Behavior learning with Max-Min Target Value Regularization
11   Generate imagined trajectories $\{(z_i, a_i)\}_{i=t}^{t+H}$ with $\pi_{\psi}^{(T)}$ and $\mathcal{M}_{\phi}^{(T)}$.
12   Compute the critic loss for $v_{\xi}^{(T)}$ over $\{(z_i, a_i)\}_{i=t}^{t+H}$ using Eq. (6) with source domain guidance.
13   Compute the actor loss for $\pi_{\psi}^{(T)}$ over $\{(z_i, a_i)\}_{i=t}^{t+H}$, and update ξ and ψ.
14   **end**
15   // Train the source domain agent
16   **for** *update step* $k_s = 1 \ldots K_2$ **do**
17   Draw a batch of trajectories from the source buffer, $\{(o_t, a_t, r_t)\}_{t=1}^{T} \sim \mathcal{B}^{(S)}$.
18   // Behavior learning with Target-Modulated Source Reward Prediction
19   Compute the target-modulated source domain rewards $\{r_t^{(S)}\}_{t=1}^{T}$ using Eq. (4).
20   Compute the reward prediction MLE loss for $\hat{r}_{\phi'}^{(S)}$ using Eq. (5).
21   Compute the dynamics loss for $\mathcal{M}_{\phi'}^{(S)}$, and update model parameters ϕ'.
22   Train the source actor $\pi_{\psi'}^{(S)}$ and critic $v_{\xi'}^{(S)}$ over the imagined $\{(z_i, a_i)\}_{i=t}^{t+H}$, and update ψ' and ξ'.
23   Append $\mathcal{B}^{(S)}$ by interacting with source environment using $\mathcal{M}_{\phi'}^{(S)}$ and $\pi_{\psi'}^{(S)}$.
24   **end**
25   **end**

3.2   Domain-Collaborative Representation Learning

The domain-collaborative representation learning process aligns the state space distributions produced by the online and offline world models. It facilitates a more generalizable world model, which in turn contributes to the transferability in value estimation.

3.2.1   Source Model Pretraining

We follow a model-based RL approach named DreamerV2 [13] to pretrain the source domain agent. The training process consists of two sub-stages: (i) world model learning using data samples drawn from the source buffer and (ii) behavior learning over latent imaginations. The world model $\mathcal{M}^{(S)}$ is primarily comprised of the following components:

$$
\begin{aligned}
&\text{Recurrent module:} && h_t = g_{\phi'}^{(S)}(h_{t-1}, z_{t-1}, a_{t-1}) &&\text{Encoder module:} && \tilde{z}_t = e_{\phi'}^{(S)}(o_t) \\
&\text{Representation module:} && z_t \sim q_{\phi'}^{(S)}(z_t \mid h_t, o_t) &&\text{Reward module:} && \hat{r}_t \sim r_{\phi'}^{(S)}(\hat{r}_t \mid h_t, z_t) \\
&\text{Transition module:} && \hat{z}_t \sim p_{\phi'}^{(S)}(\hat{z}_t \mid h_t) &&\text{Discount module:} && \hat{\gamma}_t \sim p_{\phi'}^{(S)}(\hat{\gamma}_t \mid h_t, z_t) \\
&\text{Observation module:} && \hat{o}_t \sim p_{\phi'}^{(S)}(\hat{o}_t \mid h_t, z_t)
\end{aligned}
\tag{1}
$$

where z_t is the latent state, h_t is the deterministic recurrent state. $\mathcal{M}^{(S)}$ consists of a Recurrent State-Space Model (RSSM), an image encoder, and a reward predictor. All components are implemented with neural networks, and ϕ' represents their combined parameters. In the behavior learning stage, $\mathcal{M}^{(S)}$ is used to predict future latent trajectories to train the actor and critic modules:

$$
\text{Actor: } \hat{a}_t \sim \pi_{\psi'}^{(S)}(\hat{a}_t \mid \hat{z}_t), \quad \text{Critic: } v_{\xi'}^{(S)}(\hat{z}_t) \approx \mathbb{E}_{p_{\phi'}, p_{\psi'}}\Big[\sum_{\tau \geq t} \hat{\gamma}^{\tau - t} \hat{r}_\tau\Big].
\tag{2}
$$

3.2.2 Latent Space Alignment

Utilizing the online agent to directly instruct the offline agent in learning its behavior can introduce a potential mismatch issue due to the discrepancy in tasks and dynamics between the two domains. We tackle this by incorporating the information obtained from the auxiliary domain. Concretely, we feed the same observations sampled from the target buffer $\mathcal{B}^{(T)}$ into the online and offline world models. We optimize $\mathcal{M}_\phi^{(T)}$ by

$$
\mathcal{L}_{\mathrm{wm}} \doteq \mathbb{E}_{q_\phi^{(T)}} \Big[\sum_{t=1}^{T} \underbrace{-\ln p_\phi^{(T)}(o_t \mid h_t, z_t)}_{\text{image log loss}} \underbrace{-\ln r_\phi^{(T)}(r_t \mid h_t, z_t)}_{\text{reward log loss}} \underbrace{-\ln p_\phi^{(T)}(\gamma_t \mid h_t, z_t)}_{\text{discount log loss}}
$$
$$
+ \underbrace{\beta_1 \mathrm{KL}\Big[q_\phi^{(T)}(z_t \mid h_t, o_t) \,\|\, p_\phi^{(T)}(z_t \mid h_t)\Big]}_{\text{KL loss}} + \underbrace{\beta_2 \mathrm{KL}\Big[\mathtt{sg}(e_{\phi'}^{(S)}(o_t))\|e_\phi^{(T)}(o_t)\Big]}_{\text{domain KL loss}} \Big], \tag{3}
$$

where $\mathtt{sg}(\cdot)$ indicates gradient stopping, and β_1 and β_2 are hyperparameters. We minimize the Kullback-Leibler (KL) divergence between the latent state distributions produced by $e_\phi^{(T)}$ and $e_{\phi'}^{(S)}$. The state alignment method ensures that the input state distribution of the critic network in the source domain training is similar to its input state distribution during subsequent target domain training, such that $p(v_{\xi'}^{(S)}(\hat{z}_t^{(T)}))$ can be approximated by $p(v_{\xi'}^{(S)}(\hat{z}_t^{(S)}))$. It forms the fundamental basis for the subsequent domain-collaborative behavior learning process. Another benefit of this method is that, since the source world model can interact with the online environment and gather rich information, it keeps the target world model from overfitting the offline visual dataset.

3.3 Domain-Collaborative Behavior Learning

We here discuss the details of domain-collaborative behavior learning, which consists of two main aspects: *Target-modulated source reward prediction* and *max-min target value regularization*.

3.3.1 Target-Modulated Source Reward Prediction

To enable the source domain critic to have the ability to value the target domain behaviors, it is essential to provide the auxiliary RL agent with prior knowledge of the offline task. In other words, it is necessary to encourage the source agent to explore trajectories similar to those in the target domain. To achieve this, we train the reward predictor of the source model using mixed supervision from both the online domain $\mathcal{B}^{(S)}$ and the offline dataset $\mathcal{B}^{(T)}$. With an observation $o_t^{(T)}$ sampled from $\mathcal{B}^{(T)}$, we use the source domain recurrent and representation module to extract the recurrent state and latent state, denoted by $h_t^{(T)}$ and $z_t^{(T)} \sim q_{\phi'}^{(S)}(z_t^{(T)} \mid h_t^{(T)}, o_t^{(T)})$. The source domain reward can be relabeled as:

$$
r_t^{(S)} = \begin{cases} r_t^{(S)}, & (o_t^{(S)}, r_t^{(S)}) \sim \mathcal{B}^{(S)} \\ k \cdot r_{\phi'}^{(S)}(h_t^{(T)}, z_t^{(T)}) + (1-k) \cdot r_t^{(T)}, & (o_t^{(T)}, r_t^{(T)}) \sim \mathcal{B}^{(T)} \end{cases}, \tag{4}
$$

where $r_{\phi'}^{(S)}(\cdot)$ is the source domain reward predictor, and k is a hyperparameter that controls the replay ratio between the output of $r_{\phi'}^{(S)}(\cdot)$ given $o_t^{(T)}$ and the true target reward $r_t^{(T)}$. To train $r_{\phi'}^{(S)}(\cdot)$, we use a maximum likelihood estimation (MLE) approach with the following loss function:

$$
\mathcal{L}_r = \mathbb{E}_{\mathcal{B}^{(S)}}\Big[\sum_{t=1}^{T} -\ln r_{\phi'}^{(S)}(h_t^{(S)}, z_t^{(S)})\Big] + \mathbb{E}_{\mathcal{B}^{(T)}}\Big[\sum_{t=1}^{T} -\ln r_{\phi'}^{(S)}(h_t^{(T)}, z_t^{(T)})\Big]. \tag{5}
$$

This function measures the negative log-likelihood of observing the actual reward $r_t^{(S)}$ given the latent state $z_t^{(S)}$ and recurrent state $h_t^{(S)}$. By minimizing this loss function, we can optimize the reward module to accurately predict the reward for a given state and history. Through the behavior learning on source domain imaginations (Line 22 in Alg. 1), the target-modulated reward predictor enables the source domain agent to value the target domain policy.

3.3.2 Max-Min Target Value Regularization

We use the auxiliary source critic to estimate the value function during target domain behavior learning. To avoid value overestimation or over-conservatism, we add a regularization term to the critic loss, which minimizes the maximum value between the estimates from the auxiliary critic and the target critic:

$$\mathcal{L}(\xi) \doteq \mathbb{E}\left[\sum_{t=1}^{H-1} \frac{1}{2}\left(v_\xi^{(T)}(\hat{z}_t) - \mathrm{sg}(V_t^{(T)})\right)^2 + \alpha \max\left(\zeta \cdot v_\xi^{(T)}(\hat{z}_t), \mathrm{sg}\left(v_{\xi'}^{(S)}(\hat{z}_t)\right)\right)\right], \quad (6)$$

where V_t represents the target value at time step t, which is estimated using the cumulative reward. ζ is a hyperparameter that unifies the estimation scales of values in different domains.

The loss function consists of two parts: fitting cumulative value estimates and regularizing the overestimated values for out-of-distribution (OOD) data in a mildly-conservative way. The max-min operation is designed to balance between value overestimation and over-conservatism, in the sense that we do not directly penalize the value function of state-action pairs that are out-of-dataset during behavior learning, but only when the values exceed the scope of the source value function. When the target value estimate is too large, that is, $v_\xi^{(T)}(\hat{z}_t) \geq v_{\xi'}^{(S)}(\hat{z}_t)$, the second term in Eq. (6) will minimize this value down to the target value estimation $v_\xi^{(T)}(\hat{z}_t)$. This encourages the target critic to be more conservative and avoid overestimation of the true value function. On the contrary, if the auxiliary critic yields larger values, the max-min regularization term remains constant and does not contribute to the training of the offline critic module. This approach allows the source domain to guide the optimization of the target policy without excessive conservatism.

Summary. Overall, CoWorld combines the advantages of both offline and online domains: it enables efficient learning on the offline dataset while mitigating issues such as representation overfitting and value overestimation through efficient interactions with the auxiliary source domain.

4 Experiments

In this section, we evaluate CoWorld on the well-established DeepMind Control and Meta-World benchmark, comparing to the popular model-based and model-free visual RL algorithms. CoWorld outperforms the baselines in a variety of visual RL transfer tasks. We implement CoWorld with a single NVIDIA 3090 GPU, which reaches 600K steps in under 4 days. Additional implementation details and experiment results are provided in the supplementary materials.

4.1 Experimental Setup

Benchmarks. We evaluate CoWorld on two reinforcement learning environments, *i.e.*, DeepMind Control Suite [34] and Meta-World [39].

- **DeepMind Control Suite (DMC)**. It is a popular benchmark for continuous control problems. We use the *Walker* and *Cheetah* as the base agents, select six different tasks (*i.e.*, *Walker Walk*, *Walker Downhill*, *Walker Uphill*, *Walker Nofoot*, *Cheetah Downhill*, *Cheetah Nopaw*). *Walker Nofoot* represents the task in which we cannot control the right foot of the *Walker* agent. Similarly, *Cheetah Nopaw* represents the task in which we cannot control the front paw and two joints of the *Cheetah* agent.

- **Meta-World**. It is an open-source simulated benchmark designed for solving various robot control problems. Meta-World provides a wide range of manipulation tasks that require a high degree of control and coordination of a robot arm. These tasks include pick-and-place, pushing, pressing, and other complex actions that are essential for real-world robotic applications. In our study, we choose eight robot control tasks (*Window Close*, *Button Press*, *Assembly*, etc.) as the source or target domain tasks.

Dataset setting. For the target domain dataset, the offline data is generated by DreamerV2, which is similar to the collection methods in D4RL [6]. Our experiments use the *medium-replay* quality dataset for offline algorithm training. The *medium-replay* dataset comprises all the samples in the replay buffer during the training process until the policy reaches the medium level of performance
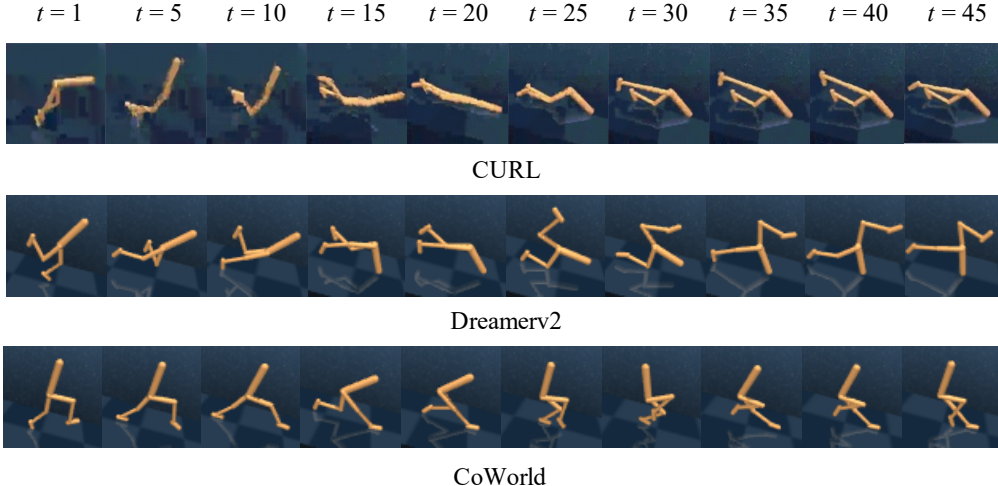
Figure 3: Visualization on *Walker Downhill*, where CoWorld demonstrates a better policy.

Table 1: Performance on DMC. We report the mean rewards and standard deviations of 10 episodes.

SOURCE TASK	TARGET TASK	CURL	CURL+CQL	DREAMERV2	COWORLD
WALKER WALK	WALKER DOWNHILL	36±25	36±5	387± 107	**511±82**
WALKER WALK	WALKER UPHILL	26±10	17±5	279±37	**344±59**
WALKER WALK	WALKER NOFOOT	25±12	34±22	309±100	**364±62**
CHEETAH RUN	CHEETAH DOWNHILL	68±7	4±1	703±52	**740±23**
CHEETAH RUN	CHEETAH NOPAW	11± 3	2±0	352±159	**465±53**

(1/3 of the maximum score that the DreamerV2 agent can achieve). To create the source domain replay buffer, we record the logging data of the agent online interacting with the environment. This data is then added to the source domain replay buffer to update the source agent parameters.

Compared methods. For the visual control tasks, we compare our approach with the following baselines, including both model-based and model-free methods, *i.e.*, DreamerV2 [13], CURL [16], and conservative Q-learning (CQL) [15].

- **DreamerV2** [13]: It is a model-based RL method and learns behaviors purely from the latent-space predictions of a separately trained world model. In order to adapt to the offline setting, we only sample the batch from the dataset and do not interact with the real environment.

- **CURL** [16]: It is a model-free RL approach that extracts high-level features from raw pixels utilizing contrastive learning. Afterward, it uses SAC [9] for control based on the extracted features.

- **CQL** [15]: It is a framework for offline RL that learns a Q-function that guarantees a lower bound for the expected policy value compared to the actual policy value. This approach effectively mitigates value function overestimation caused by distributional shifts.

4.2 DeepMind Control Suite

Implementation details. In order to verify the enhancement of CoWorld by using an auxiliary task that can be accessed online, we evaluate CoWorld on environments from DMC generalization benchmark. Source agents are trained with standard DMC environments, target agents are trained in modified DMC environments. In this modified environment, *Walker Uphill* represents the task in which the plane is a $15°$ uphill slope. *Walker Downhill* and *Cheetah Downhill* represents the task in which the plane is a $15°$ downhill slope. We evaluate our model with baselines in 5 tasks with different source domains and target domains.
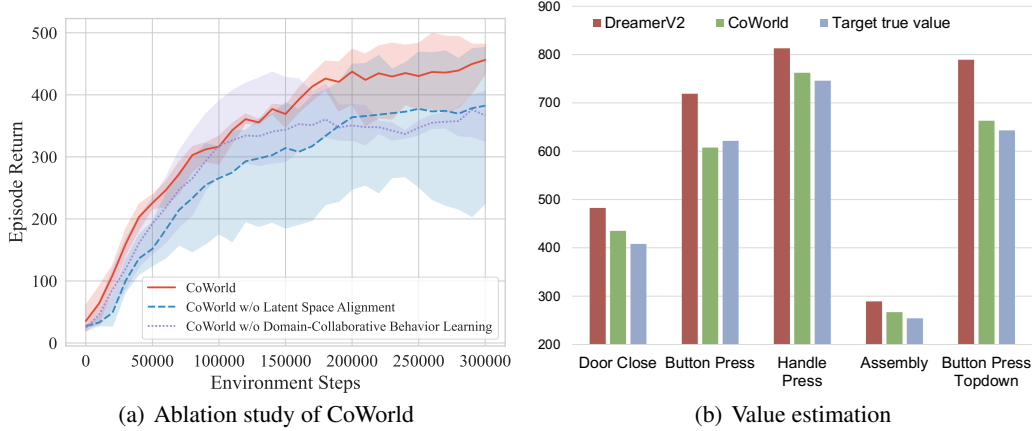
(a) Ablation study of CoWorld

(b) Value estimation

Figure 4: (a) Ablation studies that can show the effect of latent space alignment (blue) and domain-collaborative behavior learning (purple). (b) The true value of the target and the value estimations of different models in Meta-World tasks (the cumulative sum of 500 steps). The horizontal axis indicates various models that just completed training on Meta-World tasks. For better visualization, the values are rescaled. Please refer to the supplementary materials for full comparisons.

Quantitative and qualitative results. We first evaluate CoWorld on DMC, in which the tasks of the source domain and target domain are different. As shown in Table 1, CoWorld outperforms DreamerV2 and other baselines in all tasks. We evaluate the trained agent on the *Walker Downhill* task and select the first 45 frames for comparison. The visualization of *Walker Downhill* comparison is shown in Figure 3. From the evaluation results, we can find that domain-collaborative learning can effectively bridge the latent space across two domains and enable the source RL agent to provide guidance with target inclines. Moreover, ablation studies are conducted to validate the effectiveness of latent space alignment and domain-collaborative behavior learning. Figure 4(a) shows that the performance drops when domain-collaborative representation learning is removed, indicating the significance of aligning the state space distribution produced by the online and offline world models. Without the domain-collaborative behavior learning, CoWorld demonstrates lower performance since value overestimation or over-conservatism of the target domain value function.

4.3 Meta-World Benchmark

Implementation details. To explore the effectiveness of CoWorld in handling complex visual dynamics, Meta-World is used as the benchmark. Meta-World environment is widely used for evaluating the generalization and robustness of RL algorithms in simulated robotic control tasks with complex visual dynamics. For instance, the *Door Close* task require the agent to close a door with a revolving joint while randomizing the door positions, and the *Handle Press* task involves pressing a handle down while randomizing the handle positions. These tasks involve complex visual dynamics and require the agent to execute more difficult operations. To evaluate the performance of CoWorld on these environments, we compare it with several baselines in five visual RL transfer tasks.

Quantitative results. In Table 2, we present the comparative results of CoWorld and other baselines on the simulation-based experiments. Notably, CoWorld achieves the best performance in all five offline tasks compared with the baselines. Due to the OOD and overfitting issue of representation learning, directly applying the visual RL algorithms within the limited offline dataset in simulation yields poor performance in most tasks. The performance of the critic is evaluated on Meta-World tasks after the final training, and the results are presented in Figure 4(b). The value estimate is based on the cumulative critic prediction of 500 steps, while the target true value is computed by taking the discounted sum of the actual rewards obtained by the actor over the same 500 steps. The goal of the critic is to accurately estimate the target true value to avoid the misdirection of the actor caused by over-conservatism or overestimation. The experimental results indicate that DreamerV2 often overestimates values under offline conditions, while the value function of CoWorld through the online test bed assistance is more accurate. This validates the effectiveness of CoWorld.

8

Table 2: Performance on Meta-World. The agents are trained and evaluated in diverse manipulation tasks that require a robotic arm to perform various actions, such as reaching, pushing and pressing objects. We report the mean rewards and standard deviations of 10 episodes.

SOURCE TASK	TARGET TASK	CURL	CURL+CQL	DREAMERV2	COWORLD
WINDOW CLOSE	DOOR CLOSE	1972 ± 11	1984 ± 13	2785 ± 718	$\mathbf{3419 \pm 531}$
HANDLE PRESS	BUTTON TOPDOWN	189 ± 10	462 ± 67	3230 ± 999	$\mathbf{3794 \pm 74}$
DRAWER CLOSE	HANDLE PRESS	986 ± 47	988 ± 39	3404 ± 774	$\mathbf{3903 \pm 110}$
DRAWER CLOSE	BUTTON PRESS	51 ± 17	45 ± 5	3231 ± 474	$\mathbf{3623 \pm 543}$
STICK PUSH	ASSEMBLY	212 ± 15	216 ± 36	1198 ± 76	$\mathbf{1397 \pm 32}$

5 Related Work

Offline RL. The costs associated with agent interaction in real online environments are expensive and time-consuming. To address this challenge, offline RL has emerged as a promising approach, which leverages pre-collected offline data to optimize policy. However, offline RL agents are unable to interact with the online environment, which results in OOD issues [17]. There are two main categories of offline RL methods: model-free and model-based. Model-free methods focus on the conservative value estimation technique, such as CQL [15] and LAPO [3]. Notably, BCQ [7] suggests taking actions that were previously present in the offline dataset. Model-based methods (MOPO [40], COMBO [38], RAMBO [27], etc.) use an ensemble dynamic model to predict the transition process and incorporate conservative techniques to mitigate the OOD problem. Offline dataset in domains like robotic control are often available in large quantities, typically in the form of image sequences [37, 17, 30, 19, 5]. While most offline RL approaches focus on low-dimensional state problems, recent advancements [1, 26, 41] introduce offline visual RL methods that enable learning offline policies from high-dimensional image inputs.

Visual RL. Most prior works in RL focus on tasks with compact state representations. However, the ability to learn directly from rich observation spaces like images is critical for real-world applications. Therefore, many visual RL approaches are proposed. Rafailov *et al.* [26] presented an offline model-based RL algorithm that handles high-dimensional observations with latent dynamics models and uncertainty quantification. S2P (State2Pixel) [4] synthesizes the raw pixel of the agent from its corresponding state. MBRL approaches explicitly model the state transitions and generally yield higher sample efficiency than the model-free methods [29, 14]. Among them, MBRL methods based on world models [12, 11, 13, 30, 21, 23, 10, 20] show its effectiveness in visual RL.

Transfer RL. In real-life scenarios, it is essential for agents to utilize the knowledge learned in past tasks to facilitate learning in unseen tasks, which is known as transfer RL [44, 29, 42, 32, 43, 36, 8]. Sun *et al.* [33] learn a latent dynamics model in the source task and transfer the model to the target task to facilitate representation learning. APV [30] learns representations useful for understanding the dynamics of downstream domains via action-free pretraining on videos, and utilizes the pretrained representations for finetuning. The Choreographer algorithm [20] discovers, learns, and adapts skills in imagination, without the need to deploy them in the environment. VIP [18] tackles the issue of learning representations of human videos by formulating it as an offline, goal-conditioned RL problem. It introduces a self-supervised, dual goal-conditioned value-function objective that operates independently of actions, enabling pretraining utilizing unlabelled human video data. Cal-QL [22] introduces an approach for obtaining an efficient initial setup from offline data that expedites the online finetuning process.

6 Conclusions and Further Discussions

In this paper, we proposed a transfer learning method named CoWorld, which mainly tackles the difficulty in representation learning and in value estimation in offline visual RL. Our approach has two novel contributions to world model representation learning and behavior learning. First, it closes latent space across domains and mitigates overfitting problems in visual RL through domain-collaborative representation learning.

Further, it conducts domain-collaborative behavior learning with target-aware value estimation to improve offline policy performance. CoWorld achieves competitive results on the DeepMind Control Suite and the Meta-World benchmark.

It is worth noting that CoWorld follows the model-based RL baselines because the world model can serve as the simulator for policy learning and thus naturally augments the offline distributions compared with most model-free offline RL methods. However, the representation learning and behavior learning schemes in CoWorld are universal and can be readily extended to off-the-shelf model-free RL methods by additionally training a source domain reward predictor.

An unsolved problem in CoWorld is the increased computational complexity associated with the training in the auxiliary source domain. Exploring methods to improve the computational efficiency of CoWorld is a valuable direction for future research.

Acknowledgements

References

[1] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *ICML*, pages 104–114, 2020.

[2] Huayu Chen, Cheng Lu, Chengyang Ying, Hang Su, and Jun Zhu. Offline reinforcement learning via high-fidelity generative behavior modeling. In *ICLR*, 2023.

[3] Xi Chen, Ali Ghadirzadeh, Tianhe Yu, Jianhao Wang, Alex Yuan Gao, Wenzhe Li, Liang Bin, Chelsea Finn, and Chongjie Zhang. Lapo: Latent-variable advantage-weighted policy optimization for offline reinforcement learning. In *NeurIPS*, volume 35, pages 36902–36913, 2022.

[4] Daesol Cho, Dongseok Shim, and H Jin Kim. S2p: State-conditioned image synthesis for data augmentation in offline reinforcement learning. In *NeurIPS*, 2022.

[5] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. In *CoRL*, 2019.

[6] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

[7] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *ICML*, pages 2052–2062, 2019.

[8] Dibya Ghosh, Chethan Bhateja, and Sergey Levine. Reinforcement learning from passive data via latent intentions. *arXiv preprint arXiv:2304.04782*, 2023.

[9] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*, pages 1861–1870, 2018.

[10] Danijar Hafner, Kuang-Huei Lee, Ian Fischer, and Pieter Abbeel. Deep hierarchical planning from pixels. *arXiv preprint arXiv:2206.04114*, 2022.

[11] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *ICLR*, 2020.

[12] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *ICML*, pages 2555–2565, 2019.

[13] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *ICLR*, 2021.

[14] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model-based reinforcement learning for atari. In *ICLR*, 2020.

[15] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *NeurIPS*, volume 33, pages 1179–1191, 2020.

[16] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *ICML*, pages 5639–5650, 2020.

[17] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[18] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. In *ICLR*, 2023.

[19] Ajay Mandlekar, Jonathan Booher, Max Spero, Albert Tung, Anchit Gupta, Yuke Zhu, Animesh Garg, Silvio Savarese, and Li Fei-Fei. Scaling robot supervision to hundreds of hours with roboturk: Robotic manipulation dataset through human reasoning and dexterity. In *IROS*, pages 1048–1055, 2019.

[20] Pietro Mazzaglia, Tim Verbelen, Bart Dhoedt, Alexandre Lacoste, and Sai Rajeswar. Choreographer: Learning and adapting skills in imagination. In *ICLR*, 2023.

[21] Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample efficient world models. In *ICLR*, 2023.

[22] Mitsuhiko Nakamoto, Yuexiang Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. *arXiv preprint arXiv:2303.05479*, 2023.

[23] Minting Pan, Xiangming Zhu, Yunbo Wang, and Xiaokang Yang. Iso-dream: Isolating and leveraging noncontrollable visual dynamics in world models. In *NeurIPS*, volume 35, pages 23178–23191, 2022.

[24] Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Gupta. The unsurprising effectiveness of pre-trained vision models for control. In *ICML*, pages 17359–17371, 2022.

[25] Han Qi, Yi Su, Aviral Kumar, and Sergey Levine. Data-driven offline decision-making via invariant representation learning. In *NeurIPS*, 2022.

[26] Rafael Rafailov, Tianhe Yu, Aravind Rajeswaran, and Chelsea Finn. Offline reinforcement learning from images with latent space models. In *Learning for Dynamics and Control*, pages 1154–1168, 2021.

[27] Marc Rigter, Bruno Lacerda, and Nick Hawes. Rambo-rl: Robust adversarial model-based offline reinforcement learning. In *NeurIPS*, 2022.

[28] Max Schwarzer, Nitarshan Rajkumar, Michael Noukhovitch, Ankesh Anand, Laurent Charlin, R Devon Hjelm, Philip Bachman, and Aaron C Courville. Pretraining representations for data-efficient reinforcement learning. In *NeurIPS*, volume 34, pages 12686–12699, 2021.

[29] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *ICML*, pages 8583–8592, 2020.

[30] Younggyo Seo, Kimin Lee, Stephen L James, and Pieter Abbeel. Reinforcement learning with action-free pre-training from videos. In *ICML*, pages 19561–19579, 2022.

[31] Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *ICML*, pages 9870–9879, 2021.

[32] Yanchao Sun, Xiangyu Yin, and Furong Huang. Temple: Learning template of transitions for sample efficient multi-task rl. In *AAAI*, volume 35, pages 9765–9773, 2021.

[33] Yanchao Sun, Ruijie Zheng, Xiyao Wang, Andrew Cohen, and Furong Huang. Transfer rl across observation feature spaces via model-based regularization. In *ICLR*, 2022.

[34] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[35] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

[36] Mengjiao Yang and Ofir Nachum. Representation matters: offline pretraining for sequential decision making. In *ICML*, pages 11784–11794, 2021.

[37] Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Chelsea Finn, and Sergey Levine. How to leverage unlabeled data in offline reinforcement learning. In *ICML*, pages 25611–25635, 2022.

[38] Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. In *NeurIPS*, volume 34, pages 28954–28967, 2021.

[39] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *CoRL*, pages 1094–1100, 2020.

[40] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. In *NeurIPS*, volume 33, pages 14129–14142, 2020.

[41] Hongyu Zang, Xin Li, Jie Yu, Chen Liu, Riashat Islam, Remi Tachet Des Combes, and Romain Laroche. Behavior prior representation learning for offline reinforcement learning. In *ICLR*, 2023.

[42] Amy Zhang, Clare Lyle, Shagun Sodhani, Angelos Filos, Marta Kwiatkowska, Joelle Pineau, Yarin Gal, and Doina Precup. Invariant causal prediction for block mdps. In *ICML*, pages 11214–11224, 2020.

[43] Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *ICLR*, 2021.

[44] Zhuangdi Zhu, Kaixiang Lin, Anil K Jain, and Jiayu Zhou. Transfer learning in deep reinforcement learning: A survey. *arXiv preprint arXiv:2009.07888*, 2020.

[45] Zifeng Zhuang, Kun Lei, Jinxin Liu, Donglin Wang, and Yilang Guo. Behavior proximal policy optimization. In *ICLR*, 2023.

A World Model Prediction Results

The initial 5 frames are fed into the learned target domain world model of CoWorld, and this world model is applied to predict the subsequent 45 frames given actions. The results of world model prediction are shown in Figure 5.
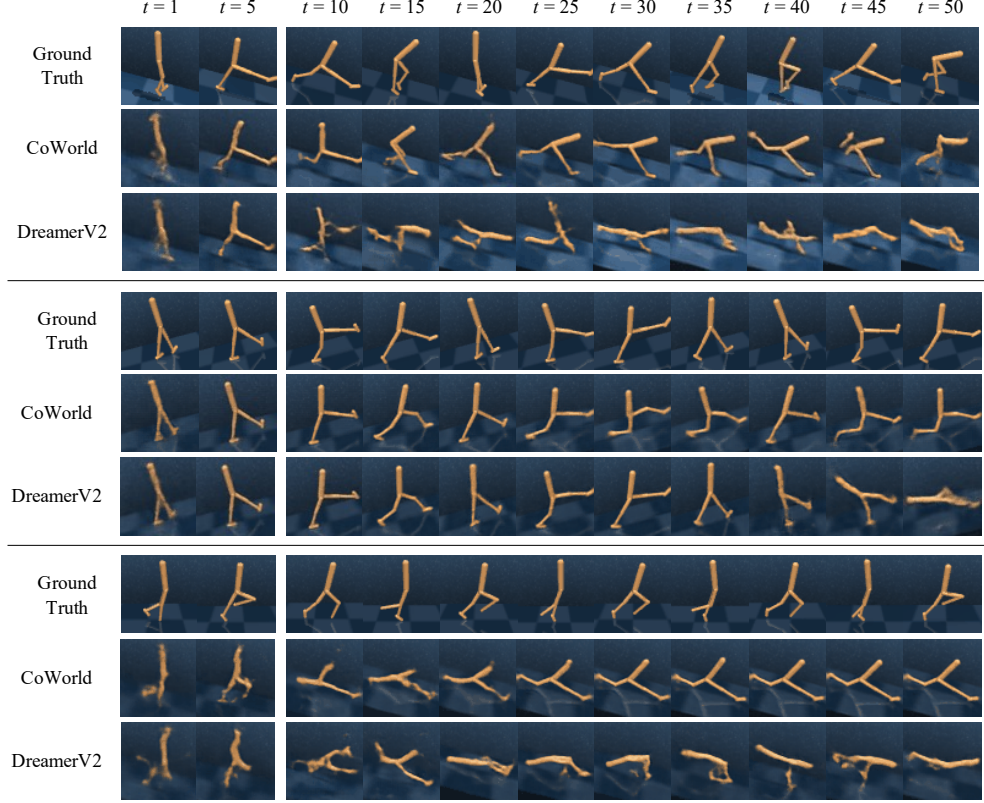

Figure 5: Visual prediction results on the DMC task *Walker Downhill* (**upper**), *Walker Uphill* (**middle**) and *Walker Nofoot* (**bottom**). For each sequence, the first 5 images are used as context frames and predict the following 45 given actions. The world model in CoWorld effectively alleviates overfitting problems in representation learning.

To investigate the impact of latent space alignment, we feed the same observation into the source and target encoder of CoWorld and then use the t-distributed stochastic neighbor embedding (t-SNE) method to visualize the latent states. As shown in Figure 6, the alignment bridges the gap between the hidden state distributions of the source encoder and target encoder.
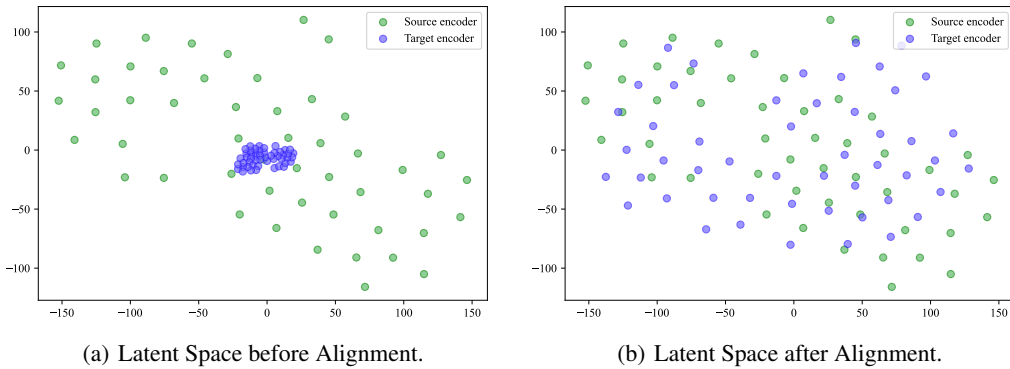

(a) Latent Space before Alignment.   (b) Latent Space after Alignment.

Figure 6: Visualization of the latent space alignment by the T-SNE method. (**a**) Latent space of CoWorld before alignment. (**b**) Latent space of CoWorld after alignment.

# B Additional Visualizations on Policy Evaluation

In Figure 7 and Figure 8, we present showcases of performing the learned policies of different models within the DMC and Meta-World domains respectively. Unlike the images in Figure 5, which are predicted by the world model, in these showcases, we directly represent the visual observations for a more meaningful comparison of the executed policies.
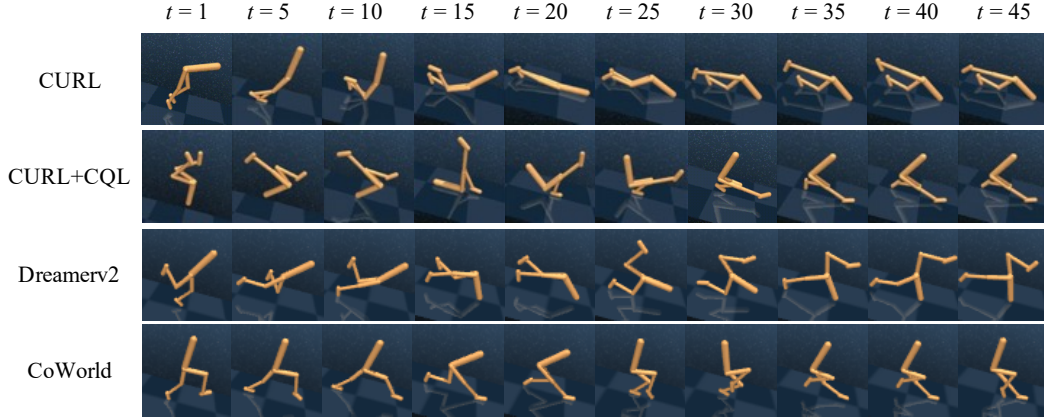
(a) Policy evaluation on the DMC *Walker Downhill* task.
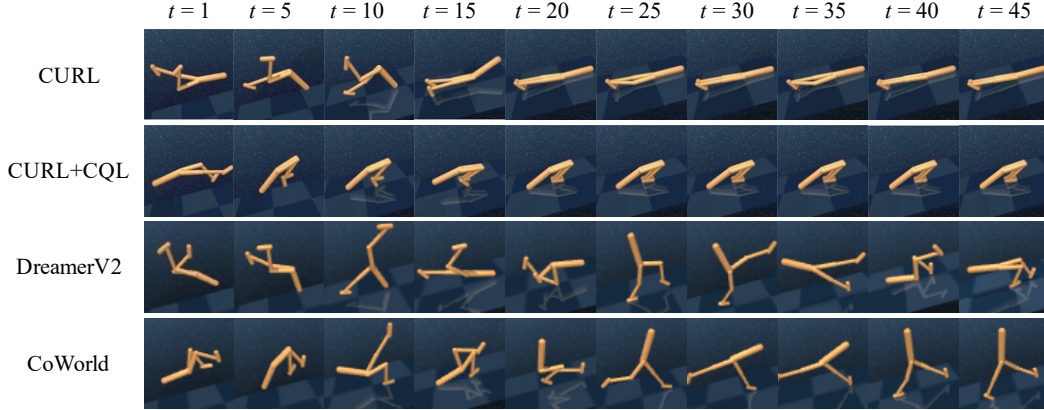
(b) Policy evaluation on the DMC *Walker Uphill* task.
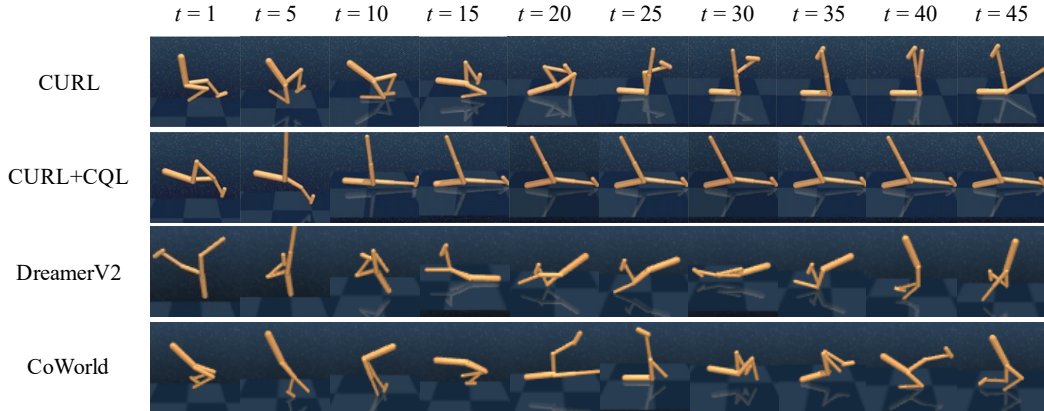
(c) Policy evaluation on the DMC *Walker Nofoot* task.

Figure 7: Additional qualitative results of policy evaluation on the DMC tasks.
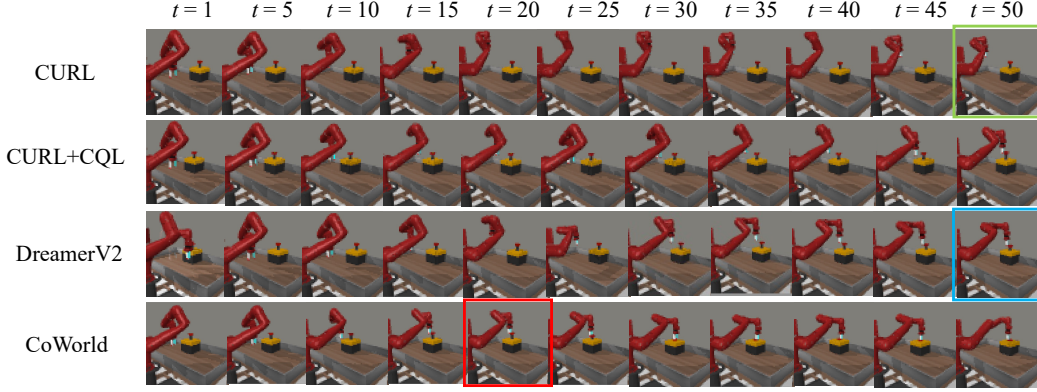
Figure 8: Policy evaluation on the Meta-World *Button Press Topdown* task. The Naive model-free method has obvious flaws and cannot complete the task (green box). Our method has better performance and completes the task in fewer steps (red box) than DreamerV2 (blue box).

(a) Comparison of CoWorld and other baselines.
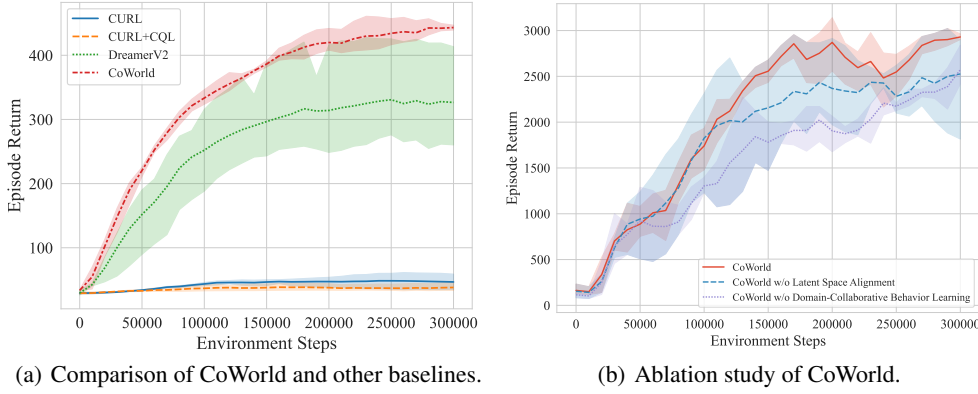
(b) Ablation study of CoWorld.

Figure 9: Performance with 3 seeds on different tasks. **(a)** Comparison of various approaches on the DMC *Walker Downhill* task, in which CoWorld outperforms DreamerV2. **(b)** Ablation studies on the Meta-World *Button Press* task that can show the impact of latent space alignment (blue) and domain-collaborative behavior learning (purple).

C   Additional Quantitative Results

Comparisons throughout the training procedure.   In Figure 9(a), we illustrate the evolution of policy returns over time on the DMC benchmark. It highlights that CoWorld exhibits notable improvements over DreamerV2 and demonstrates superior performance compared to CURL and CURL+CQL across the entire training period, in terms of both sample efficiency and final performance.

Ablation study on Meta-World benchmarks.   Figure 9(b) provides a comprehensive evaluation of different variants of CoWorld on the Meta-World Button Press task. This comparison sheds light on the individual contributions of each module incorporated in our approach to the overall performance.

Value estimation.   We compare the value estimation of CoWorld with other baselines in different tasks, including the target true value of CoWorld. As shown in Figure 10, CoWorld mitigates the overestimation problem and has better performance.

D   Hyperparameters

The hyperparameters used for CoWorld world model learning and behavior learning are shown in Table 3.
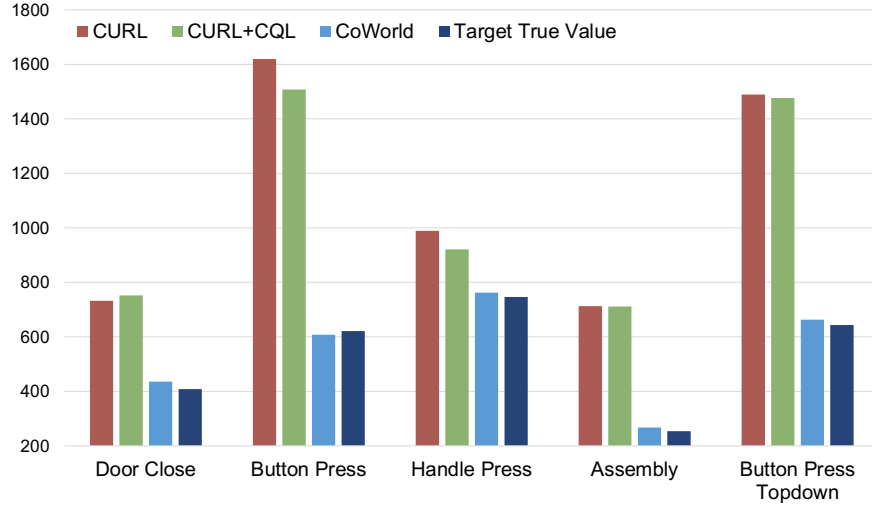
Figure 10: The true value of the target and the value estimations of different models in Meta-World tasks (the cumulative sum of 500 steps). The horizontal axis indicates various models that just completed training on Meta-World tasks. For better visualization, the values are rescaled.

Table 3: Hyperparameters of CoWorld.

Name	Notation	Value
`Co-training`		
Domain KL loss scale	β_2	1.5
Source Reward balancing	k	0.2
Target critic value loss scale	α	0.2
Source domain update steps	K_1	$2 \cdot 10^4$
Target domain update steps	K_2	$5 \cdot 10^4$
`World Model`		
Dataset size	—	$2 \cdot 10^6$
Batch size	B	50
Sequence length	L	50
KL loss scale	β_1	1
Discrete latent dimensions	—	32
Discrete latent classes	—	32
RSSM number of units	—	600
World model learning rate	—	$2 \cdot 10^{-4}$
`Behavior Leanring`		
Imagination horizon	H	15
Discount	γ	0.995
λ-target	λ	0.95
Actor learning rate	—	$4 \cdot 10^{-5}$
Critic learning rate	—	$1 \cdot 10^{-4}$