Decomposed Mutual Information Optimization for Generalized Context in Meta-Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

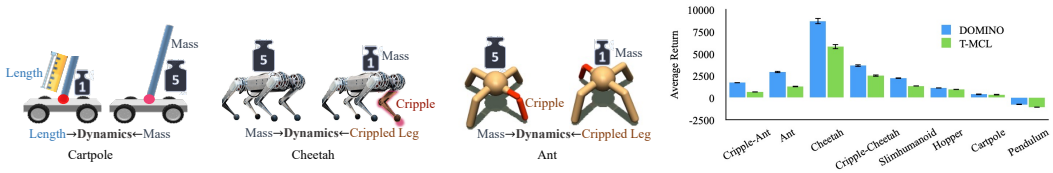
Adapting to the changes in transition dynamics is essential in robotic applications. By learning a conditional policy with a compact context, context-aware meta-reinforcement learning provides a flexible way to adjust behavior according to dynamics changes. However, in real-world applications, the agent may encounter complex dynamics changes. Multiple confounders can influence the transition dynamics, making it challenging to infer accurate context for decision-making. This paper addresses such a challenge by **D**ec**O**mposed **M**utual **IN**formation **O**ptimization (DOMINO) for context learning, which explicitly learns a disentangled context to maximize the mutual information between the context and historical trajectories, while minimizing the state transition prediction error. Our theoretical analysis shows that DOMINO can overcome the underestimation of the mutual information caused by multi-confounded challenges via learning disentangled context and reduce the demand for the number of samples collected in various environments. Extensive experiments show that the context learned by DOMINO benefits both model-based and model-free reinforcement learning algorithms for dynamics generalization in terms of sample efficiency and performance in unseen environments. Open-sourced code and videos are released on our anonymous homepage.

1 Introduction

Dynamics generalization in deep reinforcement learning (RL) investigates the problem of training a RL agent in a few kinds of environments and adapting across unseen system dynamics or structures, such as different physical parameters or robot mythologies. Meta-Reinforcement Learning (Meta-RL) has been proposed to tackle the problem by training on a range of tasks, and fast adapting to a new task with the learned prior knowledge. However, training in meta-RL requires orders of magnitudes more samples than single-task RL since the agent not only has to learn to infer the change of environment but also has to learn the corresponding policies. Context-aware meta-RL methods take a step further and show promising potential to capture local dynamics explicitly by learning an additional context vector from historical trajectories [1, 2, 3, 4]. The historical trajectories are sampled from the joint distribution of multiple confounders, which are the key factors that cause the dynamics changes. Accordingly, if multiple confounders affect the dynamics simultaneously, the state transition distribution will become highly multi-modal, leading to challenges in extracting accurate context.

Recent advanced context-aware meta-RL methods [5, 6, 7, 8] further improve meta-RL via contrastive learning, which optimizes the InfoNCE bound [9] of the mutual information in essence. These methods show a promising improvement in entangled context learning, which performs well in single confounded environments. However, as demonstrated in Figure 1a, in real-world situations for

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.



(a) Example of the multi-confounded environments (b) Performance comparison

Figure 1: Generalization with complex dynamics changes. The transition dynamics of the robot may simultaneously influenced by multiple confounders, such as mass ( ), length of leg ( ), or a crippled leg ( ). In real-world situations, all the possible confounders may change simultaneously, which bring challenges to robotic dynamics generalization. DOMINO addresses such problem by decomposed MI optimization and achieves the state-of-the-art performance.

robotic applications with partially unspecified dynamics, the transition dynamics can be influenced by multiple confounders simultaneously, such as mass changes, damping, friction, or more extreme situations with malfunctional modules like a crippled leg. For example, when a transportation robot is working in the wild, the load will dynamically change as the task progresses, while the humidity and roughness of the road also may vary. Moreover, some works also construct a confounder set for unsupervised RL environment generalization[10, 11, 12, 13, 14, 15]. Such changeable environments bring great challenges to the robot for capturing contextual information, which motivates our study.

Contribution. In this paper, we give a theoretical analysis which demonstrates that when the number of confounders increases, InfoNCE will be a loose bound of mutual information (MI) with the samples in limited seen environments, which is called MI underestimation [16]. To tackle this problem, we propose a **D**ec**O**mposed **M**utual **IN**formation **O**ptimization (DOMINO) framework for context learning in meta-RL. The context encoder aims to embed the past state-action pairs into disentangled context vectors and is optimized by maximizing the mutual information between the disentangled context vectors and historical trajectories while minimizing the state transition prediction error. DOMINO decomposes the full MI optimization problem into a summation of N smaller MI optimization problems by learning disentangled context. We then theoretically prove that DOMINO could alleviate the underestimation bias of the InfoNCE and reduce the demand for the samples collected in various environments [17, 18]. Last, with the learned disentangled context, we further develop the context-aware model-based and model-free algorithms to learn the context-conditioned policy and illustrate that DOMINO can consistently improve generalization performance in both ways to overcome the challenge of multi-confounded dynamics.

Extensive experiments demonstrate that DOMINO benefits meta-RL on both the generalization performance in unseen environments and sample efficiency during the training process under the challenging multi-confounded setting. For example, as show in Figure 1b, it achieves 1.5 times performance improvement to T-MCL [3] in the Cheetah domain and 2.6 times performance improvement to T-MCL in the Crippled-Ant domain. Visualization of the learned context demonstrates that the disentangled context generated by DOMINO under different environments could be more clearly distinguished in the embedding space, which indicates its advantage to extract high-quality contextual information from the environment.

2 Related Work

2.1 Meta-Reinforcement Learning

Meta-RL extends the framework of meta-learning [19, 20] to reinforcement learning, aiming to learn an adaptive policy being able to generalize to unseen tasks. Specifically, meta-RL methods learn the policy based on the prior knowledge discovered from various training environments and reuse the policy to fast adapt to unseen testing environments after zero or few shots. Gradient-based meta-RL algorithms [21, 22, 23, 24] learn a model initialization and adapt the parameters with few policy gradient updates in new dynamics. Context-based meta-RL algorithms [1, 2, 3, 4] learn contextual information to capture local dynamics explicitly and show great potential to tackle generalization tasks in complicated environments. Many model-free context-based methods are proposed to learn a policy conditioned on the latent context that can adapt with off-policy data by leveraging context information and is trained by maximizing the expected return. PEARL [1] adapts

to a new environment by inferring latent context variables from a small number of trajectories. Recent advanced methods further improve the quality of contextual representation leveraging contrastive learning [5, 6, 7, 8]. Unlike the model-free methods mentioned above, context-aware world models are proposed to learn the dynamics with confounders directly. CaDM [25] learns a global model that generalizes across tasks by training a latent context to capture the local dynamics. T-MCL [4] further enhances this method by combining multiple-choice learning with context-aware world model and achieves state-of-the-art results on the dynamics generalization tasks. RIA [26] introduces a relational intervention approach to T-MCL [4] which also attempts to maximize the MI between context vector and historical trajectories.

However, existing context-based approaches focus on learning entangled contexts. In a multi-confounding environment, learning entangled contexts requires orders of magnitude higher samples to capture accurate dynamics information. This is due to the highly multi-modal state transitions caused by the combined effects of multiple confounding factors. To tackle this challenge, DOMINO learns disentangled context and reduces the amount of demand for diverse samples by decomposed mutual information optimization and thus improves the generalization of the policy to overcome the adaptation problem in multi-confounded unseen environments.

2.2 Mutual Information Optimization for Representation Learning

Representation learning based on mutual information (MI) maximization has been applied in various tasks such as computer vision [27, 28], natural language processing [29, 18], and RL [30], exploiting noise-contrastive estimation (NCE) [31], InfoNCE [9] and variational objectives [32]. InfoNCE has gained recent interest with respect to variational approaches due to its lower variance [33] and superior performance in downstream tasks. However, InfoNCE may underestimate the true MI, given that it is limited by the number of samples. To tackle this problem, DEMI [16] first scaffolds the total MI estimation into a sequence of smaller estimation problems. In this paper, since the confounders in the real world are commonly independent, we simplify the complexity of mutual information decomposition and eliminate the need to learn conditional mutual information as a sub-term, assuming that multiple confounders are independent of each other.

3 Preliminaries

We consider standard RL framework where an agent optimizes a specified reward function through interacting with an environment. Formally, we formulate our problem as a Markov decision process (MDP) [34], which is defined as a tuple $(\mathcal{S}, \mathcal{A}, p, r, \gamma, \rho_0)$. Here, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $p\left(s'|s,a\right)$ is the transition dynamics, $r\left(s,a\right)$ is the reward function, ρ_0 is the initial state distribution, and $\gamma \in [0,1)$ is the discount factor. The goal of RL is to obtain a policy, mapping from states to actions, that maximizes the expected return. In order to address the problem of generalization, we further consider the distribution of MDPs, where the transition dynamics $p_{\tilde{u}}\left(s'|s,a\right)$ varies according to multiple confounders $\tilde{u} = \{u_0, u_1, \ldots, u_N\}$. The confounders can be continuous random variables, like the mass, damping, random disturbance force, or discrete random variables, such as one of the robot's leg is crippled. We assume that the true transition dynamics model is unknown, but the state transition data can be sampled by taking actions in the environment. Given a set of training setting sampled from $p(\tilde{u}_{\text{train}})$, the meta-training process learns a policy $\pi(s,c)$ that adapts to the task at hand by conditioning on the embedding of the history of past transitions, which we refer as context c. At test-time, the policy should adapt to the new MDP under the test setting $\tilde{u}_{\text{test}}$ drawn from $p(\tilde{u}_{\text{test}})$. Our goal is to learn a policy to maximizing the expected return $\mathcal{R}_{\text{train}}$ condition on the context is encoded from the sequences of current state action pairs $\{s_\tau, a_\tau, s_{\tau+1}\}_{\tau=t-H}^{t}$ in several training scenarios and enable it to perform well and achieve a high expected return $\mathcal{R}_{test}$ in test scenarios never seen before.

$$\mathcal{R}_{\#} = \mathbb{E}_{\tilde{u}\sim p(\tilde{u}_{\#})}\left[\sum_{t=0}^{\infty}\gamma^t r\left(\mathbf{s}_t, \mathbf{a}_t\right)\right], \quad a_t \sim \pi(s_t, c), \quad \# = \{\text{"train" or "test"}\} \tag{1}$$

4 Decomposed Mutual Information Optimization for Context Learning

In this section, we first provide a theoretical analysis to show why the multi-confounded environments are more challenging. We find that when the number of confounders increases, the InfoNCE will be a

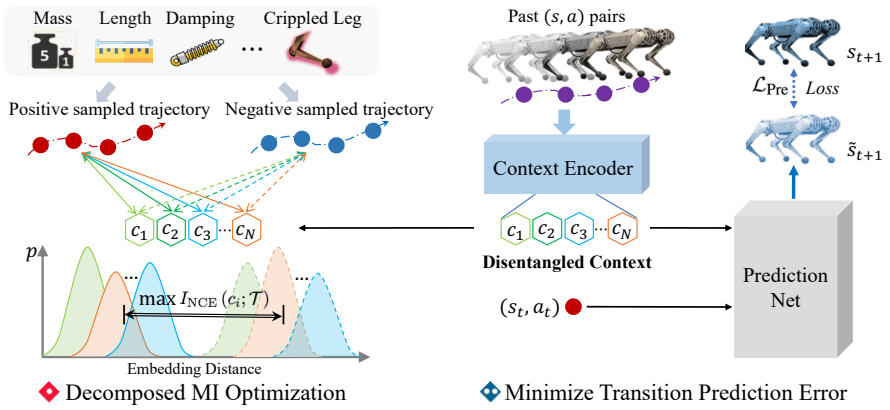


Figure 2: The overall framework of DOMINO. The context encoder embeds the past state-action pairs into disentangled context vectors. The disentangled context vectors are learned via the decomposed mutual information optimization while minimizing the state transition prediction error. **Minimize transition prediction error**: With the current state-action pair and the learned context vectors, future state can be predicted by the prediction network. The gradient of the prediction error will be used to update both the context encoder and the prediction network. **Decomposed MI Optimization**: we optimize the MI between the learned context and the historical trajectories under the same confounder setting via maximizing the InfoNCE bound, which aims to minimize the embedding distance between the positive sampled trajectories and the context, while maximizing the embedding distance between the negative sampled trajectories and the context.

loose bound of MI with the samples in limited seen environments, resulting in the underestimation of MI. To solve such a problem, we develop the DOMINO framework to learn disentangled context by decomposed MI optimization. We theoretically illustrate that the decomposed MI optimization can alleviate the underestimation of MI and reduce the demand for the number of samples. The disentangled context $c = \{c_0, c_1, \ldots, c_N\}$ is embedded by the context encoder with parameter φ from the past state-action pairs $\tau^* = \{s_l^*, a_l^*\}_{l=t-H}^{t-1}$ in current episode. DOMINO explicitly maximizes the MI between the context c and the historical trajectories $\mathcal{T} = \{\tau^i\}_{i=1}^{M}$ ($\tau^i = \{s_l^i, a_l^i\}_{l=0}^{T}$) collected based on the combination of multiple confounders $\tilde{u} = \{u_0, u_1, \ldots, u_N\}$ as same as the current confounder setting, while minimizing the state transition prediction error conditioned on the learned context. We solve the MI optimization problem by optimizing the InfoNCE lower bound on MI [9], which can be viewed as a contrastive method for the MI optimization, and decompose the full MI optimization into smaller ones to alleviate the underestimation of the mutual information and reduce the demand for the number of samples collected in various environments.

4.1 InfoNCE Bound for Mutual Information Optimization

InfoNCE bound $I_{\text{NCE}}(x;y)$ is a lower bound of the mutual information $I(x;y)$, where NCE stands for Noise-Contrastive Estimation, is a type of contrastive loss function used for self-supervised learning. InfoNCE is obtained by comparing pairs sampled from the joint distribution $x, y_1 \sim p(x,y)$ (y_1 is called the positive example) to pairs x, y_i built using a set of negative examples, $y_{2:K} \sim p(y_{2:K}) = \prod_{k=2}^{K} p(y_k)$:

$$I(x;y) \geq I_{\text{NCE}}(x;y \mid \psi, K) = E\left[\log \frac{e^{\psi(x,y_1)}}{\frac{1}{K}\sum_{k=1}^{K} e^{\psi(x,y_k)}}\right] \tag{2}$$

where ψ is a function assigning a similarity score to x, y pairs and K denotes the number of samples. Through discriminating naturally the paired positive instances from the randomly paired negative instances, it is proved to bring universal performance gains in various domains, such as computer vision and natural language processing.

Lemma 1 *$I_{\text{NCE}}(X;Y \mid K) \leq I(x;y) \leq \log K$ is a necessary condition for $I_{\text{NCE}}(X;Y \mid K)$ to be a tight bound of $I(x;y)$. (see proof in Appendix A)*

Some previous context-aware methods learn an entangled context c by maximizing the mutual information between the context c embedded from the past state-action pairs in the current episode, and the historical trajectories $\mathcal{T}$ collected under the same confounder setting as the current episode. They solve this problem by maximizing the InfoNCE lower bound on $I_{\text{NCE}}(c;\mathcal{T})$, which can be viewed as a contrastive estimation [9] of $I_{\text{NCE}}(c;\mathcal{T})$, and obtain promising improvement in single-confounded environment. However, according to Lemma 1, the $I_{\text{NCE}}(c;\mathcal{T})$ may be loose if the true mutual information $I(c;\mathcal{T})$ is larger than $\log K$, which is called underestimation of the mutual information. Therefore, to make the InfoNCE bound to be a tight bound of $I(c;\mathcal{T})$, the minimum number of samples is $e^{I(c;\mathcal{T})}$.

In real-world robotic control tasks, the dynamics of the robot is commonly influenced by multiple confounders $\tilde{u} = \{u_0, \ldots, u_i, \ldots, u_j, \ldots, u_N\}$ simultaneously, under the assumption that the confounders are independent (such as mass and damping), the mutual information between the historical trajectories $\mathcal{T}$ and the context c can be derived as

$$\begin{aligned} I(c;\mathcal{T}) &= \mathbb{E}_{p(\tau,c)}\left\{\log \frac{p(\mathcal{T} \mid c)}{p(\mathcal{T})}\right\} = \mathbb{E}_{p(\tau,c)} \log\left\{\frac{\int p(\mathcal{T} \mid \tilde{u}) p(\tilde{u} \mid c) \mathrm{d}\tilde{u}}{p(\mathcal{T})}\right\} \\ &\geq \mathbb{E}_{p(\tau,c)p(\tilde{u}|c)}\left\{\log \frac{p(\mathcal{T} \mid \tilde{u})}{p(\mathcal{T})}\right\} = I(\tilde{u};\mathcal{T}) \overset{u_i \perp u_j}{\Longrightarrow} \sum_{i=0}^{N} I\left(u_i;\mathcal{T}\right) \end{aligned} \tag{3}$$

As the number of confounders increases, the lower bound of $I(c;\mathcal{T})$ will become larger, and the necessary condition for $I_{\text{NCE}}(\mathcal{T}; c \mid K)$ to be a tight bound of $I(c;\mathcal{T})$ will become more difficult to satisfy. Since $I(c;\mathcal{T}) \geq \sum_{i=0}^{N} I\left(u_i;\mathcal{T}\right)$, to let the necessary consition satisfied, the amount of data K must be larger than $e^{\sum_{i=0}^{N} I(u_i;\mathcal{T})}$ according to Lemma 1. Thus the demand for data increases significantly. Since the confounders are commonly independent in real-world, can we relax this condition by learning disentangled context vectors instead of entangled context intuitively?

4.2 Decomposed MI Optimization

If the context vectors $c = \{c_0, c_1, \ldots, c_N\}$ can be independent, then we can ease this problem by applying the chain rule on MI to decompose the total MI into a sum of potentially smaller MI terms, i.e.,

$$I_{\text{NCE}}(c;\mathcal{T} \mid K) = \sum_{i=0}^{N} \{I_{\text{NCE}}(c_i;\mathcal{T} \mid K)\} \leq N \log K \tag{4}$$

Theorem 1 *If the context vectors $\{c_0, c_1, \ldots, c_N\}$ can be independent, then the necessary condition for $I_{\text{NCE}}(c;\mathcal{T})$ to be a tight bound can be relaxed to $I(c;\mathcal{T}) \leq N \log K = \log K^N$. Thus, the need of the number of samples can be reduced from $K \geq e^{I(c;\mathcal{T})}$ to $K \geq e^{\frac{1}{N} I(c;\mathcal{T})}$.*

Inspired by Theorem 1, we intuitively learn disentangled context vectors and maximize the mutual information between the historical trajectories $\mathcal{T}$ and the context vectors $\{c_0, \ldots, c_N\}$ while minimizing the I_{NCE} between the context vectors, i.e., to maximize the $\mathcal{L}_{\text{NCE}}$

$$\mathcal{L}_{\text{NCE}}(\varphi, w) = \sum_{i=0}^{N} I_{\text{NCE}}(c_i;\mathcal{T}) - \sum_{j=0}^{N} \sum_{i=0, i\neq j}^{N} I_{\text{NCE}}(c_i;c_j) \tag{5}$$

where the $I_{\text{NCE}}(c_i;\mathcal{T})$ can be obtained with the positive trajectory τ^+ and negative trajectories $\{\tau_k^-\}_{k=2}^{K}$, i.e.,

$$I_{\text{NCE}}\left(c_i;\mathcal{T}\right) = E\left[\log \frac{e^{\psi\left(c_i, h_w\left(\tau^+\right)\right)}}{\frac{1}{K}\left(\sum_{k=2}^{K} e^{\psi\left(c_i, h_w\left(\tau_k^-\right)\right)} + e^{\psi\left(c_i, h_w\left(\tau^+\right)\right)}\right)}\right] \tag{6}$$

The context $c = \{c_0, \ldots, c_i, \ldots, c_N\}$ ($c_i \in \mathbb{R}^m$) is encoded from the past state action pairs τ^* in current episode by $g_\varphi(\cdot)$. Both the positive trajectory (collected in same setting of the confounders) and negative trajectory (collected in different setting of the confounders) are encoded to $\mathbb{R}^m$ by $h_w(\cdot)$. The critic function $\psi(\cdot,\cdot)$ measures the cosine similarity between inputs by dot product after normalization.

Under the assumption that the setting of confounders will not change in one episode, to obtain the $I_{\text{NCE}}(c_i; c_j)$, we use c_j^+ sampled from same episode like c_i as the positive example, and use c_j^- sampled from different episode as the negative example. Thus the $I_{\text{NCE}}(c_i; c_j)$ can be derived as

$$I_{\text{NCE}}\left(c_i; c_j\right) = E\left[\log \frac{e^{\psi\left(c_i, c_j^+\right)}}{\frac{1}{K}\left(\sum_{k=2}^{K} e^{\psi\left(c_i, c_j^{k-}\right)} + e^{\psi\left(c_i, c_j^+\right)}\right)}\right] \tag{7}$$

Then, the future state s_{t+1} can be predicted with the the current state s_t, action a_t and the disentangled context vectors $\{c_{0_t}, \ldots, c_{N_t}\}$ by the state transition prediction network $f_\phi(\cdot)$. We aim to minimize the prediction loss, which is equal to maximizing

$$\mathcal{L}_{\text{Pre}}(\varphi, \phi) = E_{\tau^* \sim \mathcal{B}}\left[-\frac{1}{H}\sum_{\lambda=t}^{t+H-1} \log f_\phi\left(s_{i+1} \mid s_i, a_i, (c_{0_\lambda}, \ldots, c_{N_\lambda})\right)\right], \tau^* = \{s_l^*, a_l^*\}_{l=t-H}^{t-1} \tag{8}$$

where $\mathcal{B}$ is the training set, and H is the prediction horizon. The whole framework of DOMINO is demonstrated in Figure 2 and the overall objective function of DOMINO is

$$\mathcal{L}(\varphi, w, \phi) = \mathcal{L}_{\text{Pre}}(\varphi, \phi) + \mathcal{L}_{\text{NCE}}(\varphi, w) \tag{9}$$

4.3 Combine DOMINO with Downstream RL Methods

Combination with Model-based RL. With DOMINO we can learn the context encoder and the context-aware world model together. First, the past state-action pairs are encoded into the disentangled context vectors by the context encoder. According to the learned context, the transition prediction network predicts the future states of different actions. In particular, we use the cross entropy method (CEM) [35], a typical neural model predictive control (MPC) [36] method, to select actions, in which several candidate action sequences are iteratively sampled from a candidate distribution, which is adjusted based on best-performing action samples. The optimal action sequence $\boldsymbol{a}_{t:t+T} \doteq \{\boldsymbol{a}_t, \ldots, \boldsymbol{a}_{t+T}\}$ can be obtained by

$$\text{argmax}_{a_{t:t+T}} \sum_{\lambda=t}^{t+T} \mathbb{E}_{\tilde{f}}\left[r\left(\boldsymbol{s}_\lambda, \boldsymbol{a}_\lambda\right)\right], \quad \tilde{f} = \Pr\left(s_{t+1} \mid s_t, a_t, c_{t_0}, \ldots, c_{t_N}\right) \tag{10}$$

Then, we use the mean value of adjusted candidate distribution as action and re-plan at every timestep. We provide detailed algorithm pseudo-code in the Appendix B.1. As for the adaptation process, the policy and context encoder zero-shot adapts to the unseen confounders setting u_{test}, and we use the same adaptive planning method as T-CML[3], which selects the most accurate prediction head over a recent experience condition on the inferred context. The details is introduced in Appendix D.4

Combination with Model-free RL. Previous works show that a policy learned by model-free method can be more robust to dynamics changes when it takes the contextual information as an additional input [37, 38, 39]. Motivated by this, we investigate whether the context encoder learned by DOMINO can be used as a plug-and-play module to improve the final generalization performance of model-free RL methods. We concatenate the disentangled context encoded by a pre-trained context encoder from DOMINO and the current state-action pairs, and learn a conditional policy $\pi\left(a_t | s_t, c_0, \ldots, c_N\right)$. We use the Proximal Policy Optimization (PPO) method to train the agent [40], which learns the policy by maximizing

$$\hat{\mathbb{E}}_t\left[\frac{\pi\left(a_t \mid s_t, c_{t_0}, \ldots, c_{t_N}\right)}{\pi_{\theta_{\text{old}}}\left(a_t \mid s_t, c_{t_0}, \ldots, c_{t_N}\right)}\hat{A}_t - \beta \text{KL}\left[\pi_{\theta_{\text{old}}}\left(\cdot \mid s_t, c_{t_0}, \ldots, c_{t_N}\right), \pi\left(\cdot \mid s_t, c_0, \ldots, c_N\right)\right]\right] \tag{11}$$

where $\hat{A}_t$ is the estimation of the advantage function at timestep t. We provide detailed pseudo-code in the Appendix B.2.

5 Experiments

In this section, we evaluate the performance of our DOMINO method to answer the following questions: **(1)** Can DOMINO help the model-based RL methods overcome the multi-confounded challenges in dynamics generalization (see comparison with ablation in Figure 3 and Figure 4)? **(2)**

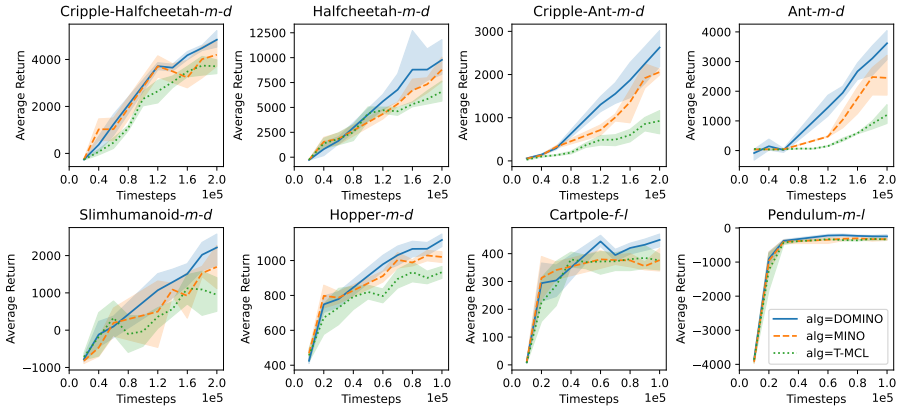


Figure 3: The average returns of the model-based methods in training environments (over 8 seeds).

Can the context encoder learned by DOMINO be used as a plug-and-play module to improve the generalization abilities of model-free RL methods in multi-confounded environments (see comparison with ablation in Table 1 and Table 2)? **(3)** Can the proposed decomposed MI optimization benefit the forward prediction of the world model? (see Figure 5) **(4)** Does the disentangled context extract more meaningful contextual information than entangled context (see Figure 6)?

5.1 Setups

We demonstrate the effectiveness of our proposed method on 8 benchmarks, which contain 6 typical robotic control tasks based on the MuJoCo physics engine [41] and 2 classical control tasks (CartPole and Pendulum) from OpenAI Gym [42]. Different from previous works, all the environments are influenced by multiple confounders simultaneously. In our experiments, we modify multiple environment parameters at the same time (e.g., mass, length, damping, push force, and crippled leg) that characterize the transition dynamics. The robotic control tasks contain 4 environments (Hopper, HalfCheetah, Ant, SlimHumanoid) affected by multiple continuous confounders and 2 more difficult environments (Crippled Ant and Crippled HalfCheetah) affected by both continuous and discrete confounders. As for Cartpole, we modify the push force f and the pole length l simultaneously. As for Pendulum, we scale the pendulum mass by scale factor m and modify the pendulum length l. As for Hopper, Half-cheetah, Ant, and Slimhumanoid, we scale the mass of every rigid link by scale factor m, and scale damping of every joint by scale factor d. As for Crippled Ant and Crippled Half-cheetah, we scale the mass of every rigid link by scale factor m, scale damping of every joint by scale factor d, and randomly select one leg, and make it crippled. The detailed settings are illustrated in Appendix C (Table 3). We implement these environments based on the publicly available code provide by [43, 4], and we also open-source the code of the multiple-confounded environments[1]. For both training and testing phase, we sample the confounders at the beginning of each episode. During training, we randomly select a combination of confounders from a training set. At test time, we evaluate each algorithm in unseen environments with confounders outside the training range.

5.2 Comparison with Model-based Methods

Baselines. We consider T-MCL [3] as the key baseline in comparison with model-based methods, which achieves the state-of-the-art results in zero-shot dynamics generalization tasks. We also consider an ***ablation version*** of DOMINO as a baseline (denoted as MINO) to show the effectiveness of the decomposed MI optimization, which optimizes the MI and predicts the future states with an entangled context without decomposition.

Results. Figure 3 shows the average return during the learning process in the training environments. The results illustrate that DOMINO learns the policy more efficiently and stably than the baselines. Figure 4 shows the generalization performance tested in the unseen environments. The results show that DOMINO surpasses T-MCL in terms of the generalization performance and the learning sample efficiency. This demonstrates that the disentangled context improves the context-aware world model. Especially, the performance gain becomes much more significant in more complex environments

[1] https://anonymous.4open.science/r/Multiple-confounded-Mujoco-Envs-01F3

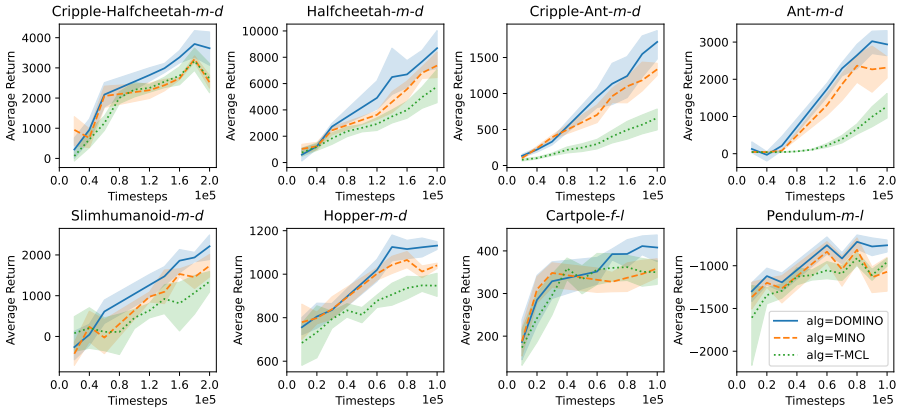


Figure 4: Comparison with *model-based methods* of the generalization performance (average return) in unseen multi-confounded environments (over 8 seeds).

(e.g., long-horizon, sensitive, and high-dimensional domains like Cripple-Ant, Ant, and Hopper). For example, DOMINO achieves about 2.6 times performance improvement to T-MCL in Cripple-Ant, which is one of the most difficult environment in the experiments, whose leg will randomly be crippled, and its mass will be changed in testing. More details are shown in the Appendix D.

5.3 Comparison with Model-free Methods

We also verify whether the learned disentangled context is useful for improving the generalization performance of model-free RL methods. Similar to [3, 44], we use the Proximal Policy Optimization (PPO [40]) method to train the agents.

Baselines. Our proposed method, which takes the context learned by DOMINO as conditional input (PPO+DOMINO), is compared with several context-conditional policies [45, 39]. Specifically, we consider combining the PPO with the context learned by T-MCL (PPO+T-MCL), which learns the context encoder via a context-aware world model and achieves the state-of-the-art performance on dynamics generalization. We also consider PEARL [45], which learns probabilistic context variable by maximizing the expected returns. We further develop an ***ablation version*** of DOMINO, which optimize the MI with entangled context (PPO+MINO) as a baseline to illustrate the effectiveness of the decomposed MI optimization. We provide more detailed explanations in Appendix D and more comparison experiments with other baselines are shown in Appendix E.2

Results. Table 1 and Table 2 show the performance of various model-free RL methods on both training and test environments. PPO+DOMINO shows superior performance and shows better generalization performances than previous conditional policy methods, implying that the proposed DOMINO method can extract contextual information more effectively than both the context learned by the model-based method (PPO+T-MCL) and the context learned by the model-free method (PEARL). Furthermore, PPO+DOMINO shows an obvious advantage over PPO+MINO, especially in complex environments, such as HalfCheetah, Ant, and Hopper, which implies that the decomposed MI optimization improves the context learning significantly. Additionally, the results also show that compared to PEARL, the context learned by T-MCL and DOMINO has better performance which implies that the state transition perdition can help to extract contextual information more effectively.

5.4 Disentangled Context Analysis

Prediction errors. To show that our method indeed helps with a transition prediction, we compare baseline methods with DOMINO in terms of prediction error across 8 environments with varying multiple confounders. As shown in Figure 5, our model demonstrates superior prediction performance, which indicates that the learned context capture better contextual information compare with entangled context (see more results in Appendix E.1).

Visualization. We visualize the disentangled context learned by DOMINO via t-SNE [46] and compare it with the entangled context learned by T-MCL. We run the learned policies under 5 randomly sampled setups of multiple confounders and collect 200 trajectories for each setting. Further, we encode the collected trajectories into context in embedding space and visualize via

Table 1: Comparison with *model-free* methods in the multi-confounded environments (over 5 seeds). The transition dynamics will change in both training and test environments in every episode.

	Cartpole-f-l		Pendulum-m-l		Ant-m-d	
	Train	Test	Train	Test	Train	Test
PEARL	197±12	175±37	-1265±173	-1293±134	153±63	73±25
PPO+T-MCL	220±27	182±25	-558±184	-579±128	176±82	173±38
PPO+MINO	267±18	234±46	-497±162	-526±219	194±95	184 ± 47
PPO+DOMINO	**299**±23	**283**±68	**-405**±139	**-436**±146	**227**±86	**216**±52
	Halfcheetah-m-d		Slimhumanoid-m-d		Hopper-m-d	
	Train	Test	Train	Test	Train	Test
PEARL	1802±773	530±270	6947±3541	3697±2674	934±242	874±366
PPO+T-MCL	2032±688	674±395	6157±1435	4136±1528	937±252	896±238
PPO+MINO	1973±563	824±498	6179±1123	4275±1134	1109±349	964±323
PPO+DOMINO	**2472**±803	**1034**±476	**7825**±1256	**5258**±1039	**1409**±254	**1137**±335

Table 2: Comparison with *model-free* methods in more difficult multi-confounded environments (over 5 seeds).

	Cripple-Ant-m-d		Cripple-Halfcheetah-m-d	
	Train	Test	Train	Test
PEARL	182±73	96 ±21	**2538**±783	1028±445
PPO+T-MCL	187±65	109±36	2368±726	1006±434
PPO+MINO	206±64	113±34	2493±664	1197±424
PPO+DOMINO	**233**±82	**132**±27	2503±658	**1326**±491

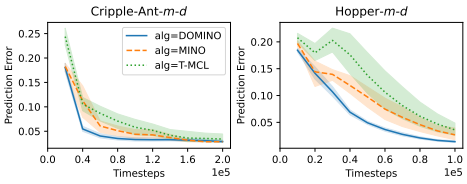
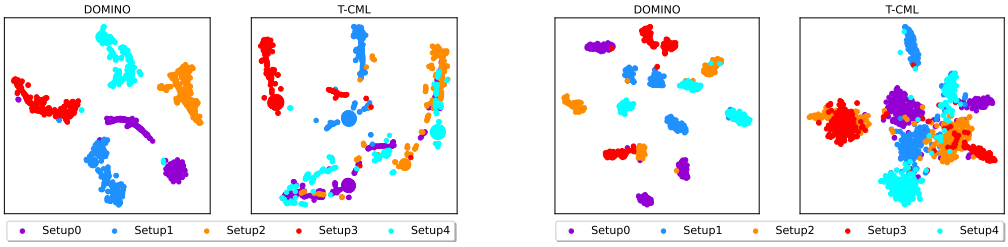


Figure 5: The testing prediction error.

t-SNE. As shown in Figure 6, we find that the disentangled context vectors encoded from trajectories collected under different confounder settings could be more clearly distinguished in the embedding space than the entangled context learned by T-MCL. This indicates that DOMINO extracts high-quality task-specific information from the environment compared with T-MCL. We provide more visualization results based on both t-SNE [46] and PCA [47] in Appendix F.2.



(a) Visualization in Hopper-m-d.

(b) Visualization in Cripple-Ant-m.

Figure 6: t-SNE visualization of context vectors extracted from trajectories collected in various environments. Embedded points from environments with the same confounders have the same color. Hopper setups: *Setup0* ($m = 0.75, d = 1.5$), *Setup1* ($m = 1.5, d = 1.25$), *Setup2* ($m = 0.5, d = 1.25$), *Setup3* ($m = 1.25, d = 0.75$), *Setup4* ($m = 1.0, d = 1.5$); Cripple-Ant setups: *Setup0* ($m = 1.15, leg = 3$), *Setup1* ($m = 0.75, leg = 1$), *Setup2* ($m = 1.25, leg = 0$), *Setup3* ($m = 0.85, leg = 0$), *Setup4* ($m = 1.0, leg = 2$).

6 Conclusion

In this paper, we propose a decomposed mutual information optimization (DOMINO) framework to learn the generalized context for zero-shot dynamics generalization. The disentangled context is learned by maximizing the mutual information between the context and historical trajectories while minimizing the state transition prediction error. By decomposing the whole mutual information optimization problem into smaller ones, DOMINO can reduce the need for samples collected in various environments and overcome the underestimation of the mutual information in multi-confounded environments. Extensive experiments illustrate that DOMINO benefits the generalization performance in unseen environments with both model-based RL and model-free RL. We believe our work can lay the foundation of dynamics generalization in complex environments.

Limitations and Negative Social Impact. DOMINO sets the number of disentangled context vectors as a hyper-parameter as same as the number of confounders in the environments. Based on our

framework, capturing the number of confounders automatically, could be future works. We believe that DOMINO will not cause any negative social impact.

References

[1] Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, pages 5331–5340, 2019.

[2] Rasool Fakoor, Pratik Chaudhari, Stefano Soatto, and Alexander J. Smola. Meta-q-learning. In *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net, 2020.

[3] Kimin Lee, Younggyo Seo, Seunghyun Lee, Honglak Lee, and Jinwoo Shin. Context-aware dynamics model for generalization in model-based reinforcement learning. In *International Conference on Machine Learning*, 2020.

[4] Younggyo Seo, Kimin Lee, Ignasi Clavera, Thanard Kurutach, Jinwoo Shin, and Pieter Abbeel. Trajectory-wise multiple choice learning for dynamics generalization in reinforcement learning. *arXiv preprint arXiv:2010.13303*, 2020.

[5] Haotian Fu, Hongyao Tang, Jianye Hao, Chen Chen, Xidong Feng, Dong Li, and Wulong Liu. Towards effective context for meta-reinforcement learning: an approach based on contrastive learning. *arXiv preprint arXiv:2009.13891*, 2020.

[6] Bernie Wang, Simon Xu, Kurt Keutzer, Yang Gao, and Bichen Wu. Improving context-based meta-reinforcement learning with self-supervised trajectory contrastive learning. *arXiv preprint arXiv:2103.06386*, 2021.

[7] Lanqing Li, Yuanhao Huang, Mingzhe Chen, Siteng Luo, Dijun Luo, and Junzhou Huang. Provably improved context-based offline meta-rl with attention and contrastive learning. *arXiv preprint arXiv:2102.10774*, 2021.

[8] Tong Sang, Hongyao Tang, Yi Ma, Jianye Hao, Yan Zheng, Zhaopeng Meng, Boyan Li, and Zhen Wang. Pandr: Fast adaptation to new environments from offline experiences via decoupling policy and environment representations. *arXiv preprint arXiv:2204.02877*, 2022.

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

[10] Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. *Advances in neural information processing systems*, 33:13049–13061, 2020.

[11] Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *International Conference on Machine Learning*, pages 4940–4950. PMLR, 2021.

[12] Minqi Jiang, Michael Dennis, Jack Parker-Holder, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Replay-guided adversarial environment design. *Advances in Neural Information Processing Systems*, 34:1884–1897, 2021.

[13] Jack Parker-Holder, Minqi Jiang, Michael Dennis, Mikayel Samvelyan, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Evolving curricula with regret-based environment design. *arXiv preprint arXiv:2203.01302*, 2022.

[14] Roberta Raileanu and Rob Fergus. Decoupling value and policy for generalization in reinforcement learning. In *International Conference on Machine Learning*, pages 8787–8798. PMLR, 2021.

[15] Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning, arxiv. *arXiv preprint arXiv:1912.01588*, 2019.

[16] Alessandro Sordoni, Nouha Dziri, Hannes Schulz, Geoff Gordon, Philip Bachman, and Remi Tachet Des Combes. Decomposed mutual information estimation for contrastive representation learning. In *International Conference on Machine Learning*, pages 9859–9869. PMLR, 2021.

[17] David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, pages 875–884. PMLR, 2020.

[18] Karl Stratos. Mutual information maximization for simple and accurate part-of-speech induction. *arXiv preprint arXiv:1804.07849*, 2018.

[19] Juergen Schmidhuber. Evolutionary principles in self-referential learning. 1987.

[20] Sebastian Thrun and Lorien Y. Pratt. Learning to learn. In *Springer US*, 1998.

[21] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, pages 1126–1135, 2017.

[22] Jonas Rothfuss, Dennis Lee, Ignasi Clavera, Tamim Asfour, and Pieter Abbeel. Promp: Proximal meta-policy search. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[23] H. Liu, R. Socher, and Caiming Xiong. Taming maml: Efficient unbiased meta-reinforcement learning. In *ICML*, 2019.

[24] A. Gupta, R. Mendonca, Yuxuan Liu, P. Abbeel, and S. Levine. Meta-reinforcement learning of structured exploration strategies. In *NeurIPS*, 2018.

[25] Kimin Lee, Younggyo Seo, Seunghyun Lee, Honglak Lee, and Jinwoo Shin. Context-aware dynamics model for generalization in model-based reinforcement learning. *CoRR*, abs/2005.06800, 2020.

[26] Jiaxian Guo, Mingming Gong, and Dacheng Tao. A relational intervention approach for unsupervised dynamics generalization in model-based reinforcement learning. In *International Conference on Learning Representations*, 2021.

[27] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[28] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[29] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[30] Bogdan Mazoure, Remi Tachet des Combes, Thang Doan, Philip Bachman, and R Devon Hjelm. Deep reinforcement and infomax learning. 2020.

[31] Michael U Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13:307–361, 2012.

[32] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. 2019.

[33] Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. 2019.

[34] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT Press, 2018.

[35] Zdravko I Botev, Dirk P Kroese, Reuven Y Rubinstein, and Pierre L'Ecuyer. The cross-entropy method for optimization. In *Handbook of statistics*. Elsevier, 2013.

[36] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

[37] Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. In *RSS*, 2018.

[38] Charles Packer, Katelyn Gao, Jernej Kos, Philipp Krähenbühl, Vladlen Koltun, and Dawn Song. Assessing generalization in deep reinforcement learning. *arXiv preprint arXiv:1810.12282*, 2018.

[39] Wenxuan Zhou, Lerrel Pinto, and Abhinav Gupta. Environment probing interaction policies. In *ICLR*, 2019.

[40] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[41] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IROS*, 2012.

[42] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[43] Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *ICLR*, 2019.

[44] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *AAAI*, 2018.

[45] Kate Rakelly, Aurick Zhou, Deirdre Quillen, Chelsea Finn, and Sergey Levine. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *ICML*, 2019.

[46] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[47] Ian T Jolliffe. *Principal component analysis for special types of data*. Springer, 2002.

[48] David Barber and Felix Agakov. The im algorithm: A variational approach to information maximization. page 201–208, 2003.

[49] Chris Cremer, Quaid Morris, and David Duvenaud. Reinterpreting importance-weighted autoencoders. 2017.

[50] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[51] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *ICLR*, 2016.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 4.2.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Section 4.2 and Appendix A.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Section 5.1 and Appendix D.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix D.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 5.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix D.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Derivations

A.1 Proof of Lemma 1

According to the Barber and Agakov's variational lower bound [48], the mutual information $I(x;y)$ between x and y can be bounded as follows:

$$I(x;y) = \mathbb{E}_{p(x,y)} \log \frac{p(y|x)}{p(y)} \geq \mathbb{E}_{p(x,y)} \log \frac{q(y|x)}{p(y)}, \tag{12}$$

where q is an arbitrary distribution. Specifically, $q(y|x)$ is defined by independently sampling a set of examples $\{y_1, \ldots, y_K\}$ from a proposal distribution $\pi(y)$ and then choosing y from $\{y_1, \ldots, y_K\}$ in proportion to the importance weights $w_y = \frac{e^{\psi(x,y)}}{\sum_k e^{\psi(x,y_k)}}$, where ψ is a function that takes x and y and outputs a scalar. According to [9], by setting the proposal distribution as the marginal distribution $\pi(y) \equiv p(y)$, the unnormalized density of y given a specific set of samples $y_{2:K} = \{y_2, \ldots, y_K\}$ and x is:

$$q\left(y \mid x, y_{2:K}\right) = p(y) \cdot \frac{K \cdot e^{\psi(x,y)}}{e^{\psi(x,y)} + \sum_{k=2}^{K} e^{\psi(x,y_k)}} \tag{13}$$

where K denotes the numbers of samples. According to [49], the expectation of $q\left(y \mid x, y_{2:K}\right)$ with respect to resampling of the alternatives $y_{2:K}$ from $p(y)$ produces a normalized density:

$$\bar{q}(y \mid x) = \mathbb{E}_{p(y_{2:K})}\left[q\left(y \mid x, y_{2:K}\right)\right] \tag{14}$$

With Equation 14 and Jensen's inequality applied in Equation 12, we have

$$\begin{aligned} I(x,y) &\geq \mathbb{E}_{p(x,y)} \log \frac{\mathbb{E}_{p(y_{2:K})} q\left(y \mid x, y_{2:K}\right)}{p(y)} \geq \mathbb{E}_{p(x,y)}\left[\mathbb{E}_{p(y_{2:K})} \log \frac{q\left(y \mid x, y_{2:K}\right)}{p(y)}\right] \\ &= \mathbb{E}_{p(x,y)}\left[\mathbb{E}_{p(y_{2:K})} \log \frac{p(y) K \cdot w_y}{p(y)}\right] \\ &= \mathbb{E}_{p(x,y)}\left[\mathbb{E}_{p(y_{2:K})} \log \frac{K \cdot e^{\psi(x,y)}}{e^{\psi(x,y)} + \sum_{k=2}^{K} e^{\psi(x,y_k)}}\right] \end{aligned} \tag{15}$$

It is obviously that $\frac{e^{\psi(x,y)}}{e^{\psi(x,y)} + \sum_{k=2}^{K} e^{\psi(x,y_k)}} \leq 1$, thus we have

$$\mathbb{E}_{p(x,y)}\left[\mathbb{E}_{p(y_{2:K})} \log \frac{K \cdot e^{\psi(x,y)}}{e^{\psi(x,y)} + \sum_{k=2}^{K} e^{\psi(x,y_k)}}\right] \leq \log K \tag{16}$$

With Equation 16, we have

$$\begin{aligned} &\mathbb{E}_{p(x,y)}\left[\mathbb{E}_{p(y_{2:K})} \log \frac{K \cdot e^{\psi(x,y)}}{e^{\psi(x,y)} + \sum_{k=2}^{K} e^{\psi(x,y_k)}}\right] \\ =&\mathbb{E}_{p(x,y_1)p(y_{2:K})}\left[\log \frac{e^{\psi(x,y)}}{\frac{1}{K}\sum_{k=1}^{K} e^{\psi(x,y_k)}}\right] = I_{\text{NCE}}(x;y \mid \psi, K) \leq \log K, \end{aligned} \tag{17}$$

Therefore, we have

$$I(x,y) \geq I_{\text{NCE}}(x;y \mid \psi, K) \leq \log K \tag{18}$$

If $I(x,y) > \log K$, then $I(x,y) > \log K \geq I_{\text{NCE}}(x;y \mid \psi, K)$, and I_{NCE} will be a loose bound. Thus, $I_{\text{NCE}} \leq I(x,y) \leq \log K$ is the necessary condition for I_{NCE} to be a tight bound of $I(x,y)$.

A.2 Detailed derivation of Theorem 1

As the number of confounders increases, although the true mutual information $I(c;\mathcal{T})$ does not increase, the necessary condition of I_{NCE} to be a tight lower bound of I_{NCE} becomes more difficult to satisfy, and the demand of data increases significantly.

As for an entangled context, the necessary condition of the InfoNCE lower bound $I_{NCE}(c;\mathcal{T})$ to be a tight bound is

$$I_{NCE}\left(c;\mathcal{T}\right) \leq I\left(c;\mathcal{T}\right) \leq \log K \tag{19}$$

Since $I(c;\mathcal{T}) \geq \sum_{i=0}^{N} I\left(u_i;\mathcal{T}\right)$, to let the above condition satisfied, the amount of data K must satisfy

$$\log K \geq \sum_{i=0}^{N} I\left(u_i;\mathcal{T}\right) \tag{20}$$

$$K \geq e^{\sum_{i=0}^{N} I(u_i;\mathcal{T})} \tag{21}$$

Therefore, if the number of confounders increases, then the demand for data will grow exponentially.

When data is not rich enough, the nesseray condition may not be satisfied. The InfoNCE lower bound $I_{NCE}(c;\mathcal{T})$ may be loose, that is $I_{NCE}(c;\mathcal{T})$ may be much smaller than the true mutual information $I(c;\mathcal{T})$, thus the MI optimization based on $I_{NCE}(c;\mathcal{T})$ will be severely affected.

$I_{NCE}\left(c_i;\mathcal{T}\right)$ is the lower bound of $I\left(c_i;\mathcal{T}\right)$ and the necessary condition of $I_{NCE}\left(c_i;\mathcal{T}\right)$ to be a tight bound of $I\left(c_i;\mathcal{T}\right)$ is

$$I_{NCE}\left(c_i;\mathcal{T}\right) \leq I\left(c_i;\mathcal{T}\right) \leq \log K \tag{22}$$

As for disentangled context $c = \{c_1, c_2, \cdots, c_N\}$, we then derive the necessary condition of $I(c,\mathcal{T})$ to be a tight lower bound of $I(c,\mathcal{T})$:

With the assumption that the contexts $\{c_1, c_2, \cdots, c_N\}$ are independent to each other, then $I(c;\mathcal{T})$ could be derived as $\sum I\left(c_i;\mathcal{T}\right)$. Therefore, under the confounder independent assumption, let $I_{NCE}(c;\mathcal{T})$ be a tight bound is only necessary to let every $I_{NCE}(c_i;\mathcal{T})$ to be a tight bound.

If every $I_{NCE}(c_i;\mathcal{T})(i = 1, 2, \ldots, N)$ is a tight bound, then we have

$$I_{NCE}\left(c_i;\mathcal{T}\right) \leq I\left(c_i;\mathcal{T}\right) \leq \log K \tag{23}$$

under the confounder independent assumption, we have

$$\sum I_{NCE}\left(c_i;\mathcal{T}\right) \leq \sum I\left(c_i;\mathcal{T}\right) \leq N \log K \tag{24}$$

$$I_{NCE}\left(c;\mathcal{T}\right) = \sum I_{NCE}\left(c_i;\mathcal{T}\right) \leq I\left(c;\mathcal{T}\right) = \sum I\left(c_i;\mathcal{T}\right) \leq N \log K \tag{25}$$

Thus, the necessary condition of $I_{NCE}\left(c;\mathcal{T}\right)$ to be a tight bound of $I\left(c;\mathcal{T}\right)$ could be relaxed to

$$I_{NCE}\left(c;\mathcal{T}\right) \leq I\left(c;\mathcal{T}\right) \leq N \log K \tag{26}$$

Therefore, by decomposing the MI estimation under the confounder independent assumption, the demand of the amount K of data could be reduced from $K \geq e^{I(c;\mathcal{T})}$ to $K \geq e^{\frac{1}{N} I(c;\mathcal{T})}$. And with $I(c;\mathcal{T}) \geq \sum_{i=0}^{N} I\left(u_i;\mathcal{T}\right)$, specificly, the the amount K of data could be reduced from $K \geq e^{\sum_{i=0}^{N} I(u_i;\mathcal{T})}$ to $K \geq e^{\frac{1}{N}\sum_{i=0}^{N} I(u_i;\mathcal{T})}$.

B Pseudo-code

B.1 Combination with model-based methods

We provide the pseudo-code of DOMINO combined with model-based methods. Firstly, the past state-action pairs are encoded into the disentangled context vectors by the context encoder. According to the

learned context, the transition prediction network predicts the future states of different actions. Then, the context encoder is optimized by maximizing the mutual information between the disentangled context vectors and historical trajectories while minimizing the state transition prediction error. In particular, we use the cross entropy method (CEM) [35], a typical neural model predictive control (MPC) [36] method, to select actions, in which several candidate action sequences are iteratively sampled from a candidate distribution, which is adjusted based on best-performing action samples.

Algorithm 1 Training DOMINO with context-aware world model

Inputs: learning rate α,maximum number of iteration P, batch size B, the number of past observations H_{past}, maxium rollout step max_step and the number of future observations H_{future}.
Initialize parameters of prediction network ϕ, context encoder φ.
Initialize replay buffer $\mathcal{D} \leftarrow \emptyset$.
for P iterations **do**
  // COLLECT TRAINING SAMPLES
  $step = 0$
  $\mathcal{V} = 0$
  while $step \leq max_step$ **do**
    Sample $u_{\mathcal{V}} \sim p_{u_{\texttt{train}}}(u)$.
    $\mathcal{V} = \mathcal{V} + 1$
    for $t = 1$ **to** TaskHorizon **do**
      $step = step + 1$
      Get context latent vectors $c_{t_0}, c_{t_1}, \ldots, c_{t_N} = g(\tau_t; \varphi), \tau_t = \{s_l, a_l\}_{l=t-H_{\text{past}}}^{t-1}$
      Collect samples $\{(s_t, a_t, s_{t+1}, r_t, \tau_t)\}$ from the environment using the planning algorithm based on CEM with the context vectors
    end for
    Update $\mathcal{D}_{u_{\mathcal{V}}} \leftarrow \mathcal{D}_{u_{\mathcal{V}}} \cup \{(s_t, a_t, s_{t+1}, r_t, \tau_t)\}$
  end while
  // UPDATE DYNAMICS MODELS AND ENCODER
  Initialize batch $\mathcal{B} \leftarrow \emptyset$.
  for $i = 1$ **to** B **do**
    sample $\mathcal{V}^*$ from [0,$\mathcal{V}_{max}$]
    Sample $\{s_t, a_t, s_{t+1}, r_t, \tau_t\}$ from $\mathcal{D}_{u_{\mathcal{V}^*}}$
    Sample positive trajectories τ^+ from $\mathcal{D}_{u_{\mathcal{V}^*}}$
    Sample negative trajectories $\{\tau_k^-\}_{k=2}^{K}$ from $\mathcal{D}_{u_{\mathcal{V}!=\mathcal{V}^*}}$
    Get context latent vectors $c_{t_0}, c_{t_1}, \ldots, c_{t_N} = g(\tau_t; \varphi)$
    Update $\mathcal{B} \leftarrow \mathcal{B} \cup \{(s_t, a_t, s_{t+1}, r_t, \tau_t)\}$
  end for
  $\mathcal{L}^{\texttt{pred}} \leftarrow E_{\tau^* \sim \mathcal{B}}\left[-\frac{1}{H}\sum_{\lambda=t}^{t+H_{\text{future}}-1} \log f_\phi\left(s_{\lambda+1} \mid s_\lambda, a_\lambda, (c_{0_\lambda}, \ldots, c_{N_\lambda})\right)\right]$
  $\mathcal{L}^{\text{NCE}} \leftarrow \sum_i^N I_{\text{NCE}}(c_i; \mathcal{T}) - \sum_j^N \sum_{i=0, i\neq j}^N I_{\text{NCE}}(c_i; c_j)$
  Update $\varphi \leftarrow \varphi - \alpha \nabla_\varphi \mathcal{L}^{\texttt{NCE}}$
  Update $\varphi \leftarrow \varphi - \alpha \nabla_\varphi \mathcal{L}^{\texttt{pred}}$
  Update $\phi \leftarrow \phi - \alpha \nabla_\phi \mathcal{L}^{\texttt{pred}}$
end for

B.2 Combination with model-free methods

We provide the pseudo-code for the combination between DOMINO and the model-free method, which uses the context encoder learned by DOMINO as a plug-and-play module to extract accurate context. We concatenate the disentangled context encoded by a pre-trained context encoder from DOMINO and the current state-action pairs, and learn a conditional policy $\pi\left(a_t|s_t, c_0, \ldots, c_N\right)$. We choose the Proximal Policy Optimization (PPO) method [40] to train the agents.

Algorithm 2 Proximal Policy Optimization with disentangled context encoder learned by DOMINO

Inputs:Maximum number of iteration P, number of actor updates M, number of critic updates B, the KL regular coefficient λ, scaling coefficient α and the learning rate β.
Initialize parameters of policy network θ, value network ξ, and context encoder φ.
for P iterations **do**
  Encode disentangled context vectors $c_{t_0}, c_{t_1}, \ldots, c_{t_N}$ by the learned context encoder $g_\varphi(\cdot)$
  Run policy π_θ for T timesteps, collecting $\{\{s_t, c_{t_0}, c_{t_1}, \ldots, c_{t_N}, a_t, r_t\}\}_{t=1}^{T}$
  Estimate advantages $\hat{A}_t = \sum_{t'>t} \gamma^{t'-t} r_{t'} - V_\xi(s_t, c_{t_0}, c_{t_1}, \ldots, c_{t_N})$
  $\pi_{\text{old}} \leftarrow \pi_\theta$
  for M updates **do**
    $J_{\text{PPO}}(\theta) \leftarrow -\left\{\sum_{t=1}^{T} \frac{\pi_\theta(a_t|s_t, c_{t_0}, c_{t_1}, \ldots, c_{t_N})}{\pi_{old}(a_t|s_t, c_{t_0}, c_{t_1}, \ldots, c_{t_N})} \hat{A}_t - \lambda \text{KL}[\pi_{old}|\pi_\theta]\right\}$
    $\theta \leftarrow \theta - \beta \nabla_\theta J_{\text{PPO}}$
  end for
  for B updates **do**
    $L_{\text{BL}}(\xi) \leftarrow \sum_{t=1}^{T} (\sum_{t'>t} \gamma^{t'-t} r_{t'} - V_\xi(s_t, c_{t_0}, c_{t_1}, \ldots, c_{t_N}))^2$
    $\xi \leftarrow \xi - \beta \nabla_\xi L_{\text{BL}}$
  end for
  if $\text{KL}[\pi_{old}|\pi_\theta] > \beta_{\text{high}} \text{KL}_{\text{target}}$ **then**
    $\lambda \leftarrow \alpha\lambda$
  else if $\text{KL}[\pi_{old}|\pi_\theta] < \beta_{\text{low}} \text{KL}_{\text{target}}$ **then**
    $\lambda \leftarrow \lambda/\alpha$
  end if
end for

C Details about the testing environments

```
def change_env(self):
        mass = np.copy(self.original_mass)
        damping = np.copy(self.original_damping)
        mass *= self.mass_scale
        damping *= self.damping_scale
        self.model.body_mass[:] = mass
        self.model.dof_damping[:] = damping
```

Listing 1: PyTorch-style pseudo-code for dynamics change based on Mujoco engine.

For CartPole environments, we use open-source implementation of CartPoleSwingUp-v2[2], which is the modified version of original CartPole environments from OpenAI Gym. The objective of CartPole task is to swing up the pole by moving a cart and keep the pole upright. For our experiments, we modify the push force f and the pole length l simultaneously. As for Pendulum, we scale the pendulum mass by scale factor m and modify the pendulum length l. For Pendulum environments, we use the open-source implementation of from the OpenAI Gym. The objective of Pendulum is to swing up the pole and keep the pole upright within 200 timesteps. We scale the pendulum mass by scale factor m and modify the pendulum length l.

As for Hopper, Half-cheetah, Ant, and Slimhumanoid, we use the environments from MuJoCo physics engine [3], and scale the mass of every rigid link by scale factor m, and scale damping of

[2] We use implementation available at https://github.com/0xangelo/gym-cartpole-swingup

[3] We use implementation available at https://github.com/iclavera/learning_to_adapt

Table 3: Environment parameters used for the multi-confounded experiments.

	Train		Test	
CartPole	$f \in$	$\{5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0\}$	$f \in$	$\{3.0, 3.5, 16.5, 17.0\}$
	$l \in$	$\{0.40, 0.45, 0.50, 0.55, 0.60\}$	$l \in$	$\{0.25, 0.30, 0.70, 0.75\}$
Pendulum	$m \in$	$\{0.75, 0.80, 0.85, 0.90, 0.95, 1.0, 1.05, 1.10, 1.15, 1.20, 1.25\}$	$m \in$	$\{0.50, 0.70, 1.30, 1.50\}$
	$l \in$	$\{0.75, 0.80, 0.85, 0.90, 0.95, 1.0, 1.05, 1.10, 1.15, 1.20, 1.25\}$	$l \in$	$\{0.50, 0.70, 1.30, 1.50\}$
Half-cheetah	$m \in$	$\{0.75, 0.85, 1.0, 1.15, 1.25\}$	$m \in$	$\{0.40, 0.50, 1.50, 1.60\}$
	$d \in$	$\{0.75, 0.85, 1.0, 1.15, 1.25\}$	$d \in$	$\{0.40, 0.50, 1.50, 1.60\}$
Ant	$m \in$	$\{0.75, 0.85, 1.0, 1.15, 1.25\}$	$m \in$	$\{0.40, 0.50, 1.50, 1.60\}$
	$d \in$	$\{0.75, 0.85, 1.0, 1.15, 1.25\}$	$d \in$	$\{0.40, 0.50, 1.50, 1.60\}$
SlimHumanoid	$m \in$	$\{0.80, 0.90, 1.0, 1.15, 1.25\}$	$m \in$	$\{0.60, 0.70, 1.50, 1.60\}$
	$d \in$	$\{0.80, 0.90, 1.0, 1.15, 1.25\}$	$d \in$	$\{0.60, 0.70, 1.50, 1.60\}$
Crippled Ant	$m \in$	$\{0.75, 0.85, 1.0, 1.15, 1.25\}$	$m \in$	$\{0.40, 0.50, 1.50, 1.60\}$
	$d \in$	$\{0.75, 0.85, 1.0, 1.15, 1.25\}$	$d \in$	$\{0.40, 0.50, 1.50, 1.60\}$
	crippled leg:	$\{0, 1, 2\}$	crippled leg:	$\{3\}$
Crippled Halfcheetah	$m \in$	$\{0.75, 0.85, 1.0, 1.15, 1.25\}$	$m \in$	$\{0.40, 0.50, 1.50, 1.60\}$
	$d \in$	$\{0.75, 0.85, 1.0, 1.15, 1.25\}$	$d \in$	$\{0.40, 0.50, 1.50, 1.60\}$
	crippled leg:	$\{0\}$	crippled leg:	$\{1\}$

```
def reset_model(self):
        c = 0.01
        self.set_state(
            self.init_qpos + self.np_random.uniform(low=-c, high=c, size=self.model.nq),
            self.init_qvel + self.np_random.uniform(low=-c, high=c, size=self.model.nv,)
        )
        pos_before = mass_center(self.model, self.sim)
        self.prev_pos = np.copy(pos_before)

        random_index = self.np_random.randint(len(self.mass_scale_set))
        self.mass_scale = self.mass_scale_set[random_index]

        random_index = self.np_random.randint(len(self.damping_scale_set))
        self.damping_scale = self.damping_scale_set[random_index]

        self.change_env()
        return self._get_obs()

```

Listing 2: PyTorch-style pseudo-code for multi-confounded environments initialization.

every joint by scale factor d. As for Crippled Ant and Crippled Half-cheetah, we scale the mass of every rigid link by scale factor m, scale damping of every joint by scale factor d, and randomly select one leg, and make it crippled. The objectives of these tasks are to move forward as fast as possible while minimizing the action cost. The detailed settings are illustrated in Table 3. We provide the pyTorch-style pseudo-code for multi-confounded environments in Listing 1 and Listing 2. We implement these environments based on the publicly available code provide by [43, 4], and we also open-source the code of the multiple-confounded environments[4]. For both the training and testing phase, we sample the confounders at the beginning of each episode. During training, we randomly select a combination of confounders from a training set. At test time, we evaluate each algorithm in

[4] We provide open-source environments at https://anonymous.4open.science/r/Multiple-confounded-Mujoco-Envs-01F3

unseen environments with confounders outside the training range. We also provide the PyTorch-style pseudo-code for the dynamics change based on Mujoco engine.

D Implementation details

D.1 Combination with Model-based RL

The context encoder is modeled as multi-layer perceptrons (MLPs) with 3 hidden layers and N output heads which are single-layer MLPs. Every disentangled context vector is produced as a 10-dimensional vector by the 3 hidden layers and a specific output head. Then, the disentangled context vectors are used as the additional input to the prediction network, i.e., the input is given as a concatenation of state, action, and context vector. We use $H_{past} = 10$ for the number of past observations and $H_{future} = 5$ for the number of future observations. The prediction network is modeled as multi-layer perceptrons (MLPs) with 4 hidden layers of 200 units each and Swish activations. For each prediction head, the mean and variance are parameterized by a single linear layer that takes the output vector of the backbone network as an input. To train the prediction network, we collect 10 trajectories with 200 timesteps from environments using the MPC controller and train the model for 50 epochs at every iteration. We train the prediction network for 10 iterations for every experiment. We evaluate trained models on environments over 8 random seeds every iteration to report the testing performance. The Adam optimizer [50] is used with a learning rate 1×10^{-4}. For planning, we use the cross entropy method (CEM) with 200 candidate actions for all the environments. The horizon of MPC is set as 30.

D.2 Combination with Model-free RL

We train the model-free agents for 5 million timesteps on OpenAI-Gym and MuJoCo environments (i.e., Hopper, Half-cheetah, Ant, Crippled Half-cheetah, Crippled Half-Ant, Slim-Humanoid) and 0.5 million timesteps on CartPole and Pendulum. The trained agents are evaluated every 10,000 timesteps over 5 random seeds. We use a discount factor $\gamma = 0.99$, a generalized advantage estimator [51] parameter $\lambda = 0.95$ and an entropy bonus of 0.01 for exploration. In every iteration, the agent rollouts 200 timesteps in the environments with the learned policy, and then it will be trained for 8 epochs with 4 mini-batches. The Adam optimizer is used with the learning rate 5×10^{-4}.

D.3 Details of InfoNCE

We provide detailed pseudocode for the calculation of InfoNCE bound. Specifically, the temperature τ is set as 0.004 to the calculation of $I(c_i, \mathcal{T})$ and is set as 0.1 to calculation of $I(c_i, c_j)$.

Algorithm 3 Pseudocode of InfoNCE in a PyTorch-like style.

```
# x_q input vector
# x_k positive sample
# x_que negative samples
# f_q, f_k, f_que: encoder networks for query, key and queue
# m: momentum
# t: temperature

def InfoNCE(x_q,x_k,x_que):
    q = f_q.forward(x_q) # queries: NxC
    k = f_k.forward(x_k) # keys: NxC
    k = k.detach() # no gradient to keys
    queue = f_que.forward(x_que) # keys: Cx(K-1)

    # positive logits: Nx1
    l_pos = bmm(q.view(N,1,C), k.view(N,C,1))

    # negative logits: Nx(K-1)
    l_neg = mm(q.view(N,C), queue.view(C,K-1))

    # logits: NxK
    logits = cat([l_pos, l_neg], dim=1)

    # contrastive loss, Eqn.(1)
    labels = zeros(N) # positives are the 0-th
    loss = CrossEntropyLoss(logits/t, labels)
```

D.4 Details of the adaptive planning used in adaption process

The prediction model has 3 output head $head_0, head_1, head_2$ which are used for selecting actions by planning. The adaptive planning method selects the most accurate prediction head over a recent experience. Given N past transitions, we select the prediction head h_* by

$$\underset{head \in [Head]}{\operatorname{argmin}} \sum_{i=t-N}^{t-2} \ell\left(s_{i+1}, f\left(s_{i+1} \mid s_i, a_i, (c_{0_\lambda}, \ldots, c_{N_\lambda}); \phi, head\right)\right)$$

where ℓ is the mean square error function. All the hyper-parameter is set as same as T-MCL[3].

E Additional Results

E.1 Prediction Error

As shown in Figure 7, DOMINO has a smaller prediction error compared to T-MCL and its ablation version MINO (optimize MI with entangled context), indicating that the learned context can effectively help predict the future state more accurately, which is the key to the performance of the model-based planning.

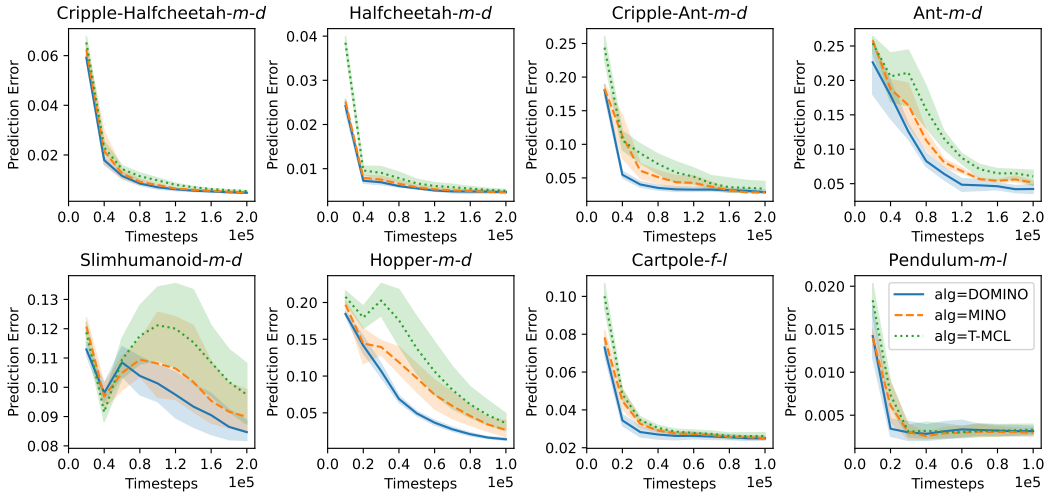


Figure 7: Comparison with the *model-based methods* of the Prediction Error. The results show the mean and standard deviation of average returns averaged over 8 runs.

E.2 Comparison with RIA

We compare DOMINO with RIA and TMCL together under multi-confounded setting, the environments including Hopper-m-d, Halfcheetah-m-d, Slim-humanoid-m-d and Pendulum-m-l. As shown in Figure 8, DOMINO achieves better generalization performance than RIA and TMCL, especially in complex environments like Halfcheetah-m-d and Slim-humanoid-m-d.

E.3 Sensitivity Analysis of the hyper-parameter N

We compare the performance of DOMINO with different hyper-parameter N, which is equal or not equal to the number of confounders in the environment. In this experiment, the confounder is the damping, mass, and a crippled leg (number of confounders is 3), and we compare the performance of DOMINO with different hyper-parameter $N = 1, 2, 3$. As shown in Figure 9, even though the hyper-parameter N is not equal to the ground truth value of the confounder number, DOMINO also benefits the context learning. In practice, under a conservative setup of hyper-parameter N, DOMINO can also benefit the context learning compared to the baselines like TMCL.

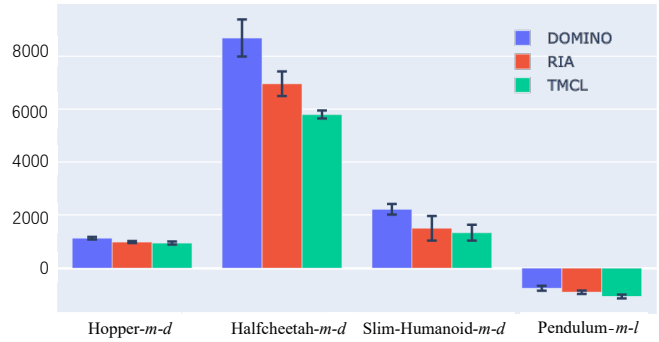
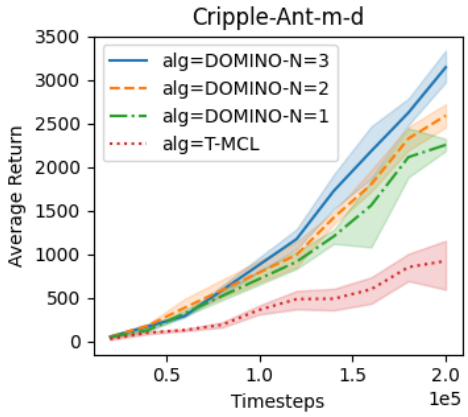
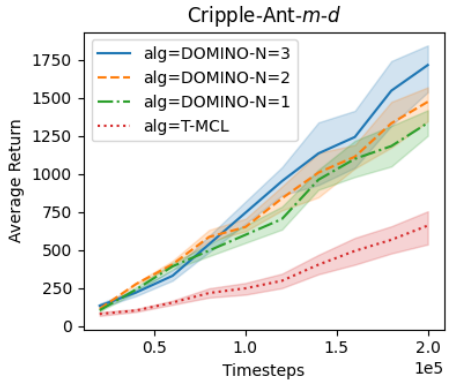


Figure 8: Generalization performance comparison between DOMINO, RIA and TMCL over 4 runs.



(a) Performance evaluation in seen environments



(b) Performance evaluation in unseen environments

Figure 9: The ablation of different N in Crippled-Ant-m-d domain(contains 3 confounders).

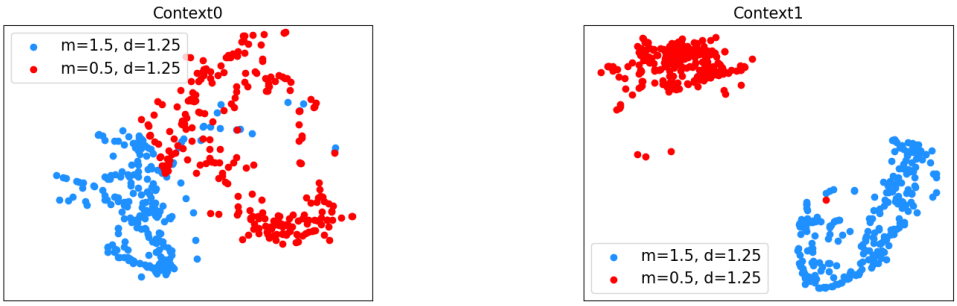
F Visualization

F.1 Verifying whether the contexts is disentangled

We add an additional experiment to show that the context vectors inferred by DOMINO are disentangled well. We vary only one of the confounders and observe the changes of N disentangled vectors. In this experiment, we set up two different confounders: mass m and damping d. Under the DOMINO framework, the context encoder inferred two disentangled context vectors: context 0 and context 1. As shown in Figure 11 and Figure 10, the context 1 is more related to damping. When the confounders are set as the same mass but different damping, the visualization result of context 1 under different settings are separated clearly from each other, while under the same damping but different mass settings, the visualization result of context 1 is much more blurred from each other. Similarly, context 0 is more related to mass. When the confounders are set to the same damping but different mass, the visualization result of context 0 under different settings is separated clearly from each other, while under the same mass but different damping settings, the visualization result of context 0 is less different from each other.

F.2 Visualization of the whole context

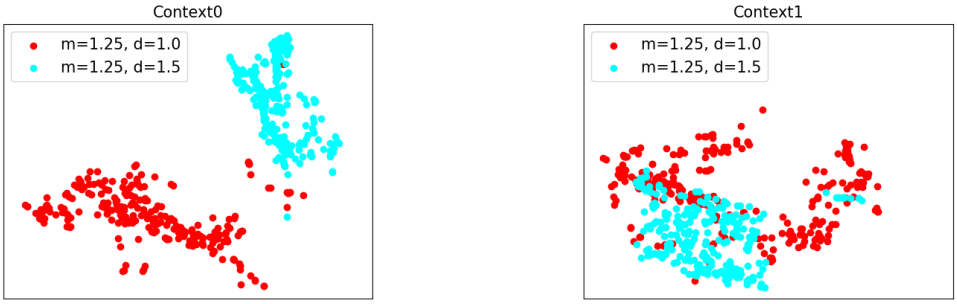
Visualization. We visualize the whole context which is a the concatenation of the disentangled contexts learned by DOMINO via t-SNE [46] and compare it with the entangled context learned by T-MCL. We run the learned policies under 5 randomly sampled setups of multiple confounders



(a) Visualization of Context0

(b) Visualization of Context1

Figure 10: Visualization of disentangled context in with same damping scale $d = 1.25$ and different mass scale $m = 1.5, m = 0.5$.

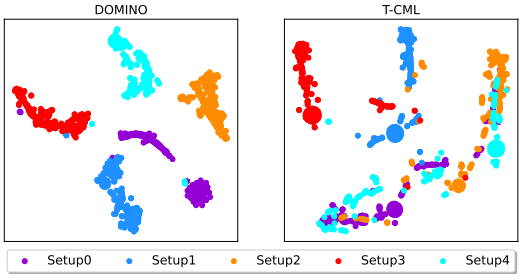


(a) Visualization of Context0

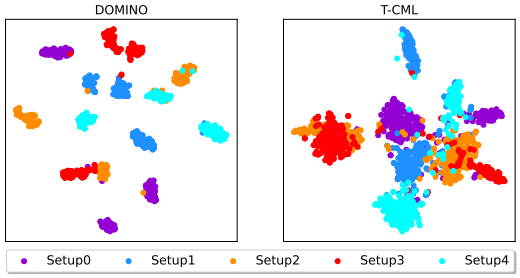
(b) Visualization of Context1

Figure 11: Visualization of disentangled context in with same mass scale $m = 1.25$ and different damping scale $d = 1.0, d = 1.5$.

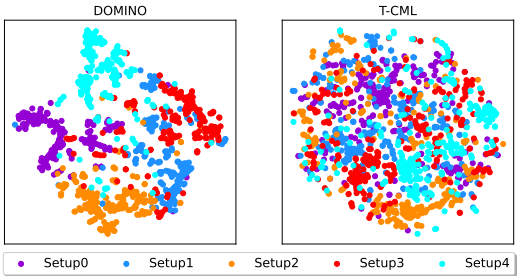
and collect 200 trajectories for each setting. Further, we encode the collected trajectories into context in embedding space and visualize via t-SNE [46] and PCA [47]. As shown in Figure 12 and Figure 13, we find that the disentangled context vectors encoded from trajectories collected under different confounder settings could be more clearly distinguished in the embedding space than the entangled context learned by T-MCL. This indicates that DOMINO extracts high-quality task-specific information from the environment compared with T-MCL. Accordingly, the policy conditioned on the disentangled context is more likely to get a higher expected return on dynamics generalization tasks, which is consistent with our prior empirical findings.



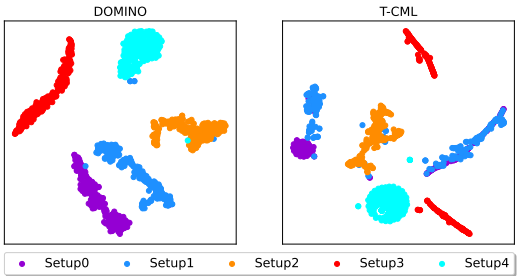
(a) Visualization in Hopper-m-d



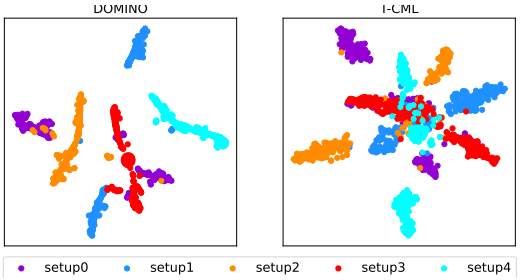
(b) Visualization in Cripple-Ant-m-d



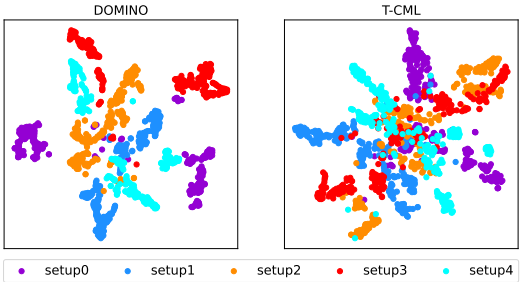
(c) Visualization in Slim-Humanoid-m-d



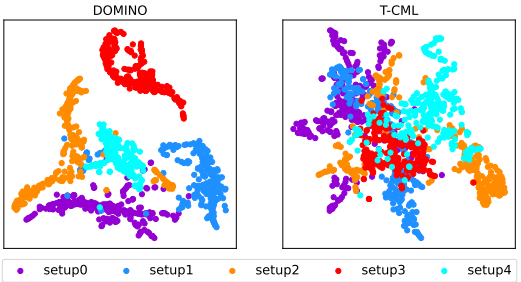
(d) Visualization in Halfcheetah-m-d



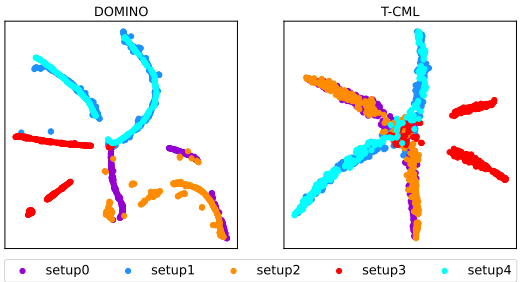
(e) Visualization in Ant-m-d



(f) Visualization in Cartpole-f-l

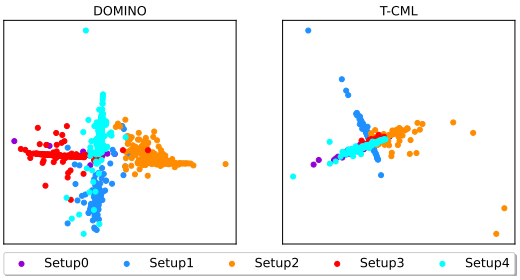


(g) Visualization in Pendulum-m-l

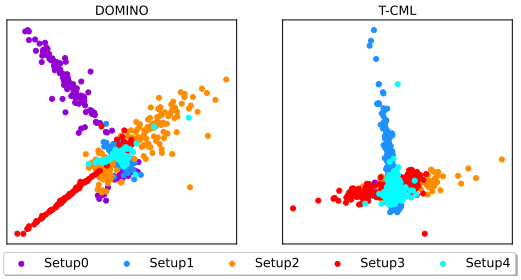


(h) Visualization in Cripple-Halfcheetah-m-d

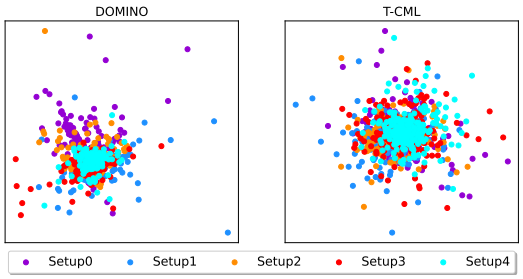
Figure 12: t-SNE [46] visualization of context vectors extracted from trajectories collected in various environments. Embedded points from environments with the same confounders have the same color.



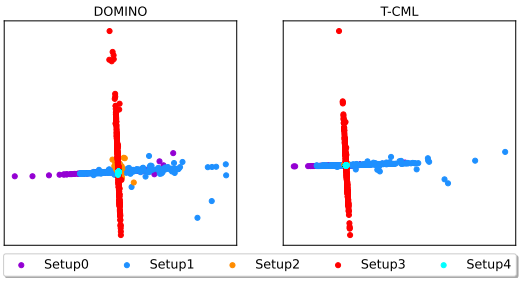
(a) Visualization in Hopper-m-d



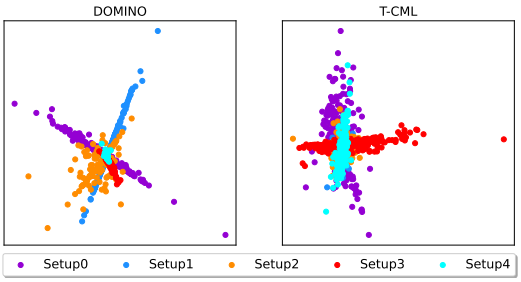
(b) Visualization in Cripple-Ant-m-d



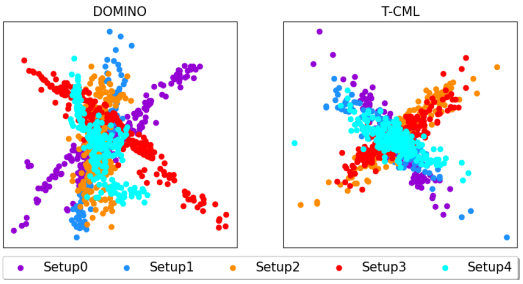
(c) Visualization in Slim-Humanoid-m-d



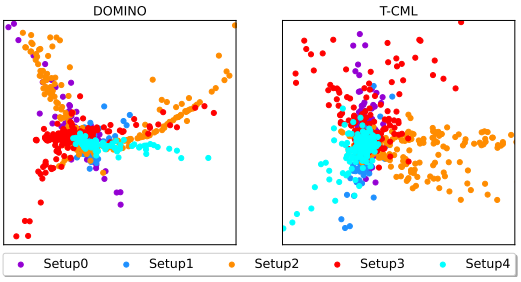
(d) Visualization in Halfcheetah-m-d



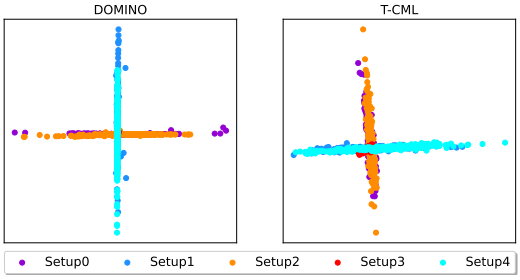
(e) Visualization in Ant-m-d



(f) Visualization in Cartpole-f-l



(g) Visualization in Pendulum-m-l



(h) Visualization in Cripple-Halfcheetah-m-d

Figure 13: PCA [46] visualization of context vectors extracted from trajectories collected in various environments. Embedded points from environments with the same confounders have the same color.