Learning Contextual Causal Dynamics for Robust Exploration in Reinforcement Learning

Jiaming Pu[1], **Yibo Zhang**[1], **Xu Dong**[1], **Dengpeng Xing**[1,2,*], **Tielin Zhang**[3,4*]

[1] Institute of Automation, Chinese Academy of Sciences
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
[3] Center for Excellence in Brain Science and Intelligence Technology
Chinese Academy of Sciences
[4] State Key Laboratory of Brain Cognition and Brain-Inspired Intelligence Technology
{pujiaming2024, zhangyibo2023, dongxu2024, dengpeng.xing}@ia.ac.cn;
{zhangtielin}@ion.ac.cn

Abstract

Efficient exploration in sparse-reward environments, where informative extrinsic feedback is scarce, remains a significant challenge in reinforcement learning. A key limitation of existing exploration strategies is that they often assume that all state variables strongly influence one another, overlooking the underlying causal mechanisms of the environment, particularly the fact that causal relationships may vary across contexts. To address this issue, we propose contextual **C**ausal dynamics model-based **I**ntrinsic **M**otivation (**CIM**), a causality-aware exploration framework that explicitly captures context-dependent sparse causal structures and derives intrinsic motivation signals, specifically *causal action influence* and *prediction-error-based curiosity*. These intrinsic rewards encourage interventions on causally relevant state factors rather than merely promoting observation novelty-seeking behavior, thereby facilitating robust exploration and active skill acquisition. Moreover, by learning contextual causal mechanisms, CIM identifies and filters out task-irrelevant state variables, improving generalization under distribution shifts. Empirical results demonstrate that our method achieves superior sample efficiency, training robustness, and out-of-distribution generalization performance across multiple robotic manipulation tasks.

1 Introduction

Efficient exploration in sparse-reward environments remains a significant challenge that limits sample efficiency in reinforcement learning (RL) [1]. In scenarios with extremely sparse feedback, where informative signals are available only after long horizons or under restrictive conditions, agents often fail to obtain sufficiently dense and discriminative learning signals, which in turn severely impedes reliable policy optimization and convergence. Intrinsic motivation-based reward shaping offers a principled framework to address this issue. Representative methodologies generally include count-based approaches [2, 3], which explicitly track state visitation frequencies to incentivize the exploration of rarely visited regions, and curiosity-driven strategies [4, 5], which utilize the prediction error of a learned dynamics model as an intrinsic reward to guide agents toward unfamiliar regions. Many of these methods are built upon learned environment dynamics, yet they often overlook the underlying mechanisms governing interactions among state variables, implicitly assuming densely entangled dependencies. As a result, they fail to provide fine-grained reward signals. Moreover, such

*Co-corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

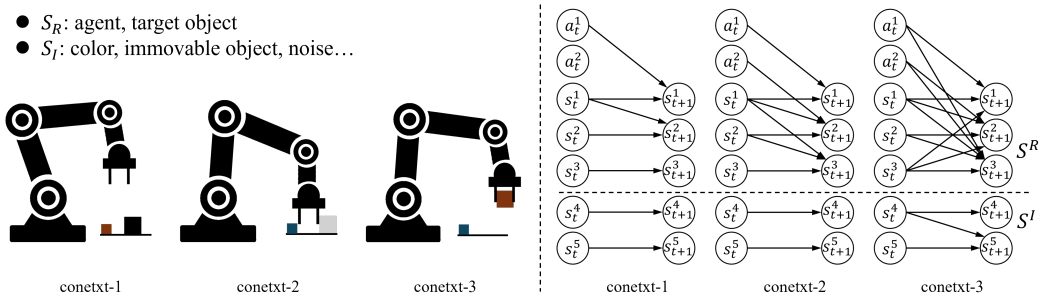


Figure 1: **Left:** Under different interaction scenarios, the full state variables influence one another through distinct causal mechanisms. Here, S_R denotes the task-relevant state variables that can be influenced by actions, including agent-related dynamic features and the target object-related dynamic features. In contrast, S_I denotes the task-irrelevant state variables that cannot be influenced by actions under any context, such as object color, noise, and immovable objects. **Right:** The multiple causal graphs represent the corresponding local causal structures under different contexts, each associated with an interaction scenario shown on the left.

densely coupled dynamics struggle to distinguish mere observational novelty from truly meaningful signals, leaving agents unable to differentiate informative state transitions from stochastic noise and causing exploration to be distracted by task-irrelevant variables. In downstream tasks, the irregular state fluctuations induced by these irrelevant variables can directly interfere with the agent's action selection.

Inspired by recent research in causal reinforcement learning [1, 6, 7], we aim to leverage a causal perspective to address these limitations of existing intrinsic-motivation methods. The objective is to empower agents with explicit causal awareness by constructing a causal dynamics model that captures the underlying causal mechanisms. However, prior causal-modeling approaches [8–10] often assume a spatially and temporally invariant global causal graph, an assumption that is often overly idealized for complex real-world environments, as demonstrated by [11]. As illustrated in Fig. 1 (left), in a robotic manipulation task the intrinsic causal mechanisms vary across interaction phases (before contact, during contact, and grasping). We refer to such context-dependent influence patterns as contextual causal relationships. Here, a context is a region of the state–action space with a locally stable causal mechanism. Modeling causality as context-dependent [12, 13] is pivotal both for accurately modeling environment dynamics and for generating meaningful intrinsic rewards that are informative and aligned with task progress.

To this end, we propose contextual **C**ausal dynamics model–based **I**ntrinsic **M**otivation (**CIM**). By learning a Contextual Causal Dynamics Model (CCDM), CIM explicitly captures locally stable causal relationships among state variables across different contexts, enabling accurate modeling of sparse and context-dependent dynamics. Crucially, the CCDM identifies and filters out task-irrelevant state variables (e.g., immovable objects, object color, random noise) as shown in Fig. 1 (left), which may induce observational novelty yet carry little meaningful information for task completion. Leveraging the CCDM, we derive fine-grained, robust-to-noise intrinsic rewards, specifically *causal action influence* and *prediction-error-based curiosity*, to facilitate effective exploration. Simultaneously, during policy learning, filtering out these task-irrelevant state variables stabilizes in-distribution training and enables generalization to out-of-distribution (OOD). Our contributions are summarized as follows:

- We introduce **CIM**, a novel framework that derives intrinsic rewards by learning a CCDM, enabling sample-efficient and robust exploration in sparse-reward environments.
- We further exploit the learned CCDM to identify and filter out task-irrelevant state variables, which helps the agent focus on causally relevant information and improves OOD generalization.
- Extensive experiments on multiple sparse-reward robotic manipulation benchmarks show that our method consistently outperforms representative baselines in sample efficiency, training robustness, and OOD generalization performance.

2 Related Work

2.1 Causal Dynamics Learning

Causality has recently been widely adopted in reinforcement learning [1, 14–17], among which a significant application of causality in RL is dynamics learning. Unlike conventional approaches [18, 19] that use dense neural networks to approximate transition dynamics, several recent studies [8, 20] advocate modeling the environment with explicit causal structure. By leveraging causal discovery techniques such as sparsity constraints [21], conditional mutual information [8], and conditional independence tests [20], these methods seek to identify true causal dependencies between states and actions. This process removes spurious correlations and facilitates the construction of accurate environmental dynamics models. Prior work has applied causal dynamics to enable planning via imagined rollouts [14, 22], support counterfactual reasoning for enhancing interpretability [9, 10], and improve robustness to non-stationarity [23, 24]. A common limitation, however, is the assumption of a single, globally invariant causal graph, a premise that often breaks down in complex real-world settings. To address this, recent research has moved toward fine-grained causal mechanisms [25, 26] introduce sample-level local causal graphs, inferring a specific graph for each transition to enable data augmentation via structural swapping, but the high computational cost of sample-level inference makes it impractical for building efficient forward dynamics models. Alternatively, fine-grained causal dynamics learning (FCDL) [12] argues that realistic environments typically decompose into distinct causal mechanisms that remain locally stable within particular state and action subspaces. While this insight informs our dynamics modeling, FCDL is mainly designed for model-based RL and relies heavily on data collected by random policies, which limits its applicability to complex manipulation tasks. Despite these advances, existing methods have yet to adequately integrate contextual causal dynamics learning with exploration strategies tailored for sparse-reward environments, and this gap motivates the present work.

2.2 Intrinsic Motivation in RL

Intrinsic motivation has emerged as a pivotal mechanism to address sparse reward problems in reinforcement learning by supplementing environmental signals with self-supervised exploration bonuses. Generally, these approaches can be categorized into count-based methods and curiosity-driven strategies [27]. Count-based methods evolved from tracking tabular visitation frequencies [28] to estimating pseudo-counts in continuous domains [2, 3, 29–32]. This is achieved through density estimation models like Context Tree Switching [2] and PixelCNN [33, 34], or by discretizing state spaces via hashing functions [35] and leveraging successor representations to capture structural reachability [36, 37]. In parallel, curiosity-driven approaches encourage agents to visit unanticipated states by maximizing the prediction error of forward dynamics models [4, 5, 38]. To mitigate the stochastic noise issue often referred to as the "Noisy-TV" problem, researchers proposed predicting in learned feature spaces [4] or utilizing random network distillation to measure novelty against fixed target networks [39–43].

However, these methods often overlook the underlying causal mechanisms when constructing environment models, making them susceptible to noise interference and struggling to provide precise and fine-grained rewards.

3 Preliminary

3.1 Markov Decision Process

We consider a reinforcement learning problem formulated as a Markov Decision Process (MDP), defined by the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma, \rho_0 \rangle$. Here, $\mathcal{S} \subseteq \mathbb{R}^{d_\mathcal{S}}$ denotes the continuous state space and $\mathcal{A} \subseteq \mathbb{R}^{d_\mathcal{A}}$ denotes the continuous action space, where $d_\mathcal{S}$ and $d_\mathcal{A}$ denote the state and action dimensions, respectively. The environment dynamics $\mathcal{P}(s'|s, a)$ specifies the probability of transitioning to state s' given the current state s and action a. The reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ provides a reward signal at each step, and $\gamma \in [0, 1)$ is the discount factor. Finally, ρ_0 denotes the distribution of the initial state s_0. The goal of the agent is to learn a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the expected

cumulative discounted reward:

$$J(\pi) = \mathbb{E}_{s_t \sim \mathcal{P}, s_0 \sim \rho_0, a_t \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]. \tag{1}$$

3.2 Structural Causal Model

A Structural Causal Model (SCM) [44] provides a formal framework for representing causal mechanisms. An SCM is typically defined as a tuple $\langle \mathcal{U}, \mathcal{V}, \mathcal{F}, P(\mathcal{U}) \rangle$ associated with a Directed Acyclic Graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here, $\mathcal{V} = \{v_1, \ldots, v_m\}$ denotes the set of observed endogenous variables, while $\mathcal{U} = \{u_1, \ldots, u_m\}$ denotes the set of unobserved exogenous variables that capture stochastic uncertainty. In $\mathcal{G}$, each directed edge $v_j \to v_i$ indicates that v_j is a direct cause of v_i. The value of each endogenous variable v_i is generated by a structural assignment function $v_i := f_i(\mathrm{PA}(v_i; \mathcal{G}), u_i)$, where $\mathrm{PA}(v_i; \mathcal{G}) \subseteq \mathcal{V} \setminus \{v_i\}$ denotes the set of parent variables of v_i in $\mathcal{G}$. Under the assumptions described in Appendix A.1–A.2, the joint distribution over $\mathcal{V}$ factorizes according to the causal graph as

$$p(v_1, \ldots, v_m) = \prod_{i=1}^{m} p(v_i \mid \mathrm{PA}(v_i; \mathcal{G})). \tag{2}$$

3.3 Local Causal Graph

Formally, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the global causal graph. Given a specific conetxt $\mathcal{D}$, the Local Causal Graph (LCG) is defined as a subgraph $\mathcal{G}_\mathcal{D} = (\mathcal{V}_\mathcal{D}, \mathcal{E}_\mathcal{D})$. Here, $\mathcal{V}_\mathcal{D} \subseteq \mathcal{V}$ represents the subset of variables that are relevant within context $\mathcal{D}$, and $\mathcal{E}_\mathcal{D} \subseteq \mathcal{E}$ denotes the set of directed edges (causal mechanisms) that are active under this context. Conceptually, LCGs can be categorized into two primary paradigms: *Sample-specific LCGs*, which characterize causal mechanisms at the finest granularity, where the causal structure is determined uniquely for each individual example; and *Subspace-specific LCGs*, which identify invariant causal patterns that remain stable across specific temporal intervals or subspaces of the state-action space, capturing shared mechanisms within a context. In this work, our contextual causal dynamics model is built upon the concept of subspace-specific local causal graphs.

4 Methodology

Overview. We model environment dynamics with a Contextual Causal Dynamics Model in a principled way that captures discrete, context-dependent causal mechanisms (Sec. 4.1). The learned dynamics is used to estimate intrinsic rewards, *Causal Action Influence* (CAI) reward and *Intrinsic Curiosity Module* (ICM) reward to drive the agent learning (Sec. 4.2).

4.1 Contextual Causal Dynamics Model

VQ-VAE. To obtain local causal graphs and model environment dynamics, we employ the Vector Quantization-Variational Autoencoder (VQ-VAE) architecture proposed in [12, 45] to identify discrete latent contexts, from which the corresponding local causal graphs can be decoded. Here, a latent context denotes a local region of the state-action space, not an additional input variable or a signal inferred from the history. For a fixed encoder and codebook, each sample (s, a) is deterministically assigned to a code $z(s, a)$. Specifically, we use an encoder E_ϕ to map each sample (s, a) into a continuous embedding $h = E_\phi(s, a)$. The embedding is then quantized to the nearest prototype in a learnable codebook $\mathcal{C} = \{e_1, \ldots, e_{K_b}\}$, where K_b denotes the codebook size and the number of learned regions. The selected prototype is obtained as

$$e = e_z, \quad \text{where } z = \arg \min_{j \in \{1, \ldots, K_b\}} \|h - e_j\|_2^2. \tag{3}$$

To recover the causal structure associated with each context, we employ a decoder D_ϕ that maps the discrete code e_z to the parameters of a Bernoulli distribution over the adjacency matrix $M_z \in \{0, 1\}^{(d_\mathcal{S} + d_\mathcal{A}) \times d_\mathcal{S}}$, where M_z denotes the local causal graph under context e_z. To enable end-to-end training while sampling discrete binary edges, we utilize the Gumbel-Softmax reparameterization trick [46]. More details can be found in Appendix A.6 and A.7.

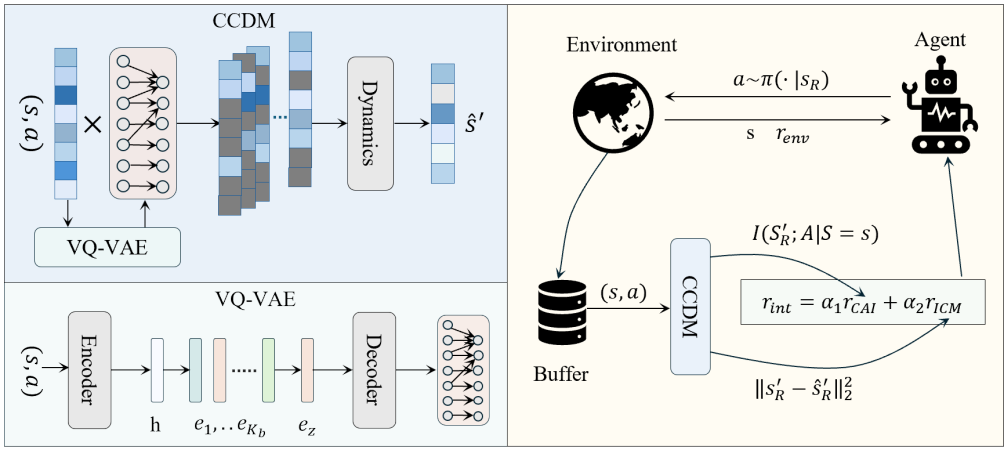


Figure 2: Overall framework. For each state–action sample (s, a), vector quantization is applied to retrieve the nearest codebook entry and decode the associated local causal graph (**VQ-VAE**). The inferred causal graph is then used to mask (s, a) along state-specific dimensions for dynamics prediction (**CCDM**). From the learned dynamics we compute intrinsic rewards r_{ICM} and r_{CAI}, which encourage the agent to interact with the environment and thus produce informative data that further enhances dynamics learning. S_R can be derived from the local causal graphs (see Sec. 4.1). For full details of the VQ-VAE architecture and the r_{CAI} calculation, see Appendix A.7 and B.

Adaptive Codebook Fusion. After each CCDM optimization and codebook EMA update, we compare code vectors using Euclidean distance, $d_z(i, j) = \|e_i - e_j\|_2$, and merge a pair when $d_z(i, j) < 2 \times 10^{-3}$. The merge is usage-weighted, and the freed entry is reinitialized for subsequent assignments; the procedure is detailed in Appendix C.1.

Causal Dynamics Prediction. The VQ-VAE maps the current state-action pair (s, a) to a code $z = z(s, a)$, which selects the corresponding local causal graph M_z. Given this graph, we factorize the next-state distribution across state dimensions, with each dimension conditioned only on its direct parents in the selected graph. This hard graph selection can be written as

$$p(s' \mid s, a; M_z) = \sum_{k=1}^{K_b} \mathbb{I}[z(s, a) = k] \prod_{i=1}^{d_{\mathcal{S}}} p(s'_i \mid \text{PA}(s'_i; M_k)), \tag{4}$$

Here, $\mathbb{I}[z(s, a) = k]$ equals 1 for the code selected by vector quantization and 0 for all other codes. Thus, exactly one term in the sum is active: the expression routes the sample to one local graph rather than averaging predictions across graphs. In each term, M_k is the graph decoded from code e_k, and $\text{PA}(s'_i; M_k)$ denotes the set of current state and action variables that are direct parents of next-state dimension s'_i in that graph. The active term therefore gives the factorized next-state prediction under the graph selected for (s, a).

Task-Relevant and Task-Irrelevant State. Based on the learned context-stable local causal graphs, we define the *Task-Irrelevant State* S_I, where

$$I = \{i \in \{1, \ldots, d_{\mathcal{S}}\} \mid \sum_{k=1}^{K_b} \sum_{j=1}^{d_{\mathcal{A}}} M_k(j, i) = 0\}, \tag{5}$$

as the set of state dimensions that receive no causal influence from actions under any context-specific causal graph M_k. These dimensions typically correspond to static attributes of environmental entities, immovable objects, background noise, and other nuisance variables. Correspondingly, the *Task-Relevant State* is defined as the complement of S_I, i.e., $S_R = S \setminus S_I$, representing the dynamic attribute dimensions of the environment state. Such a state representation enables the subsequent reward and policy to focus on more meaningful components of the state.

4.2 Causality-Aware Intrinsic Motivation

Based on the learned environment dynamics, we introduce two intrinsic rewards to encourage active skill acquisition and effective exploration of the state space. These rewards focus exclusively on the *Task-Relevant State* S_R.

Active Interaction: S_R-Based CAI. *Causal Action Influence* measures the influence and controllability of an action over the environment, namely, how much the action affects the next-state transition under the current state. Mathematically, we formulate CAI as the conditional mutual information [47] between the action A and the next task-relevant state S'_R, conditioned on the current state $S = s$:

$$r_{\text{CAI}} = I(S'_R; A \mid S = s). \tag{6}$$

Crucially, unlike prior approaches that rely on dense dynamics models, we compute r_{CAI} using the CCDM described in Sec. 4.1. Eq. 6 can be expanded as follows:

$$\begin{aligned} r_{\text{CAI}} &= \mathbb{E}_{a\sim\mathcal{U}(\mathcal{A})}\left[D_{\text{KL}}\Big(\mathcal{P}(s'_R \mid s, a; M_z) \;\Big\|\; \mathcal{P}(s'_R \mid s)\Big)\right] \\ &= \mathbb{E}_{a\sim\mathcal{U}(\mathcal{A})}\mathbb{E}_{s'_R\sim p}\left[\log \frac{p(s'_R \mid s, a; M_z)}{\int p(s'_R \mid s, a'; M_z)\, u_{\mathcal{A}}(a')\, \mathrm{d}a'}\right]. \end{aligned} \tag{7}$$

A high r_{CAI} score indicates that actions substantially influence the next task-relevant state given the current state. Here, $\mathcal{U}(\mathcal{A})$ is a fixed uniform distribution over the action space, and $u_{\mathcal{A}}$ is its density. This reference distribution is independent of the learned policy (and its goal input), so CAI measures action influence under a common intervention distribution rather than the policy's current action preferences. Driven by this reward, the agent is encouraged to explore state regions associated with behaviorally meaningful interactions while remaining robust to task-irrelevant state dimensions. In practice, we estimate r_{CAI} using Monte Carlo sampling. Detailed derivations and computation procedures are provided in Appendix B.

Auxiliary Exploration: S_R-Based ICM. We further incorporate the *Intrinsic Curiosity Module* [4] to facilitate comprehensive coverage of the state-action space, thereby accelerating exploration toward novel regions. Specifically, the module employs the forward dynamics model to predict the next task-relevant state $\hat{s}'_R$ according to Eq. 4. The intrinsic reward is defined as the corresponding prediction error:

$$r_{\text{ICM}} = \|\hat{s}'_R - s'_R\|_2^2. \tag{8}$$

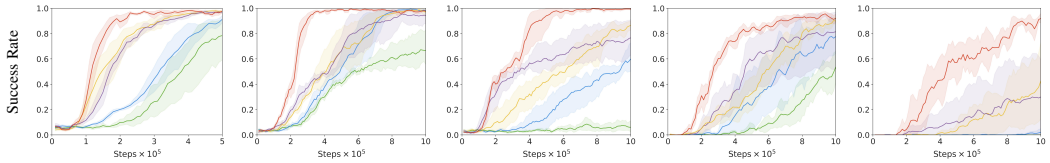
4.3 Training and Task Learning

In this section, we integrate the above components into a unified active learning framework, as illustrated in Fig. 2, to learn a dynamics model and enable the agent to accomplish tasks. The complete algorithm is presented in Algorithm 1.

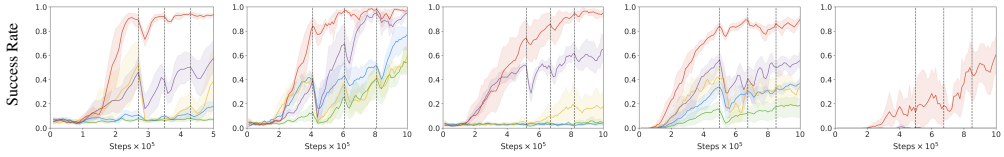
CCDM Training Optimization Objective. The entire model is trained end-to-end in an iterative loop: an RL agent collects interaction data to train the CCDM, the learned CCDM produces intrinsic rewards, and these rewards drive agent learning. The CCDM training objective $\mathcal{L}_{\text{CCDM}}$ is a weighted sum of the following components:

$$\mathcal{L}_{\text{CCDM}} = \mathcal{L}_{nll} + \beta\mathcal{L}_{vq} + \lambda\mathcal{L}_{sparse}. \tag{9}$$

Here, the Negative Log-Likelihood ($\mathcal{L}_{nll}$) measures the prediction error, formulated as $\mathcal{L}_{nll} = -\log p(s' \mid s, a; M)$. Then, the Quantization Loss ($\mathcal{L}_{vq}$) ensures the encoder commits to the discrete codebook vectors to stabilize context identification, defined as $\mathcal{L}_{vq} = \|h - \text{sg}[e_z]\|_2^2$, where $\text{sg}[\cdot]$ is the stop-gradient operator, and β is a hyperparameter. Note that we employ the Exponential Moving Average (EMA) [45] instead of direct gradient-based training to update the codebook, thereby avoiding collapse; see Appendix A.7 for details. Finally, Sparsity Regularization ($\mathcal{L}_{sparse}$) imposes an ℓ_1 penalty with a hyperparameter λ on the generated local causal graphs $M_{1,\dots,K_b}$, expressed as $\mathcal{L}_{sparse} = \|M_{1,\dots,K_b}\|_1$. This encourages the discovery of a minimal sufficient causal structure and prevents trivial fully connected solutions, thereby enabling the identification of S_R and S_I.



(a) Base performance experiments without task-irrelevant state variables.



(b) Training robustness experiments under periodic distributional shifts. The vertical dashed lines indicate each distribution shift.



Figure 3: Experiments (a), (b) are all conducted across five sparse-reward environments. From left to right, the environments are: GymFetch (Push, PickAndPlace, RotTable) and Meta-World (*pick-place*, *pick-place-wall*). Results are averaged over five random seeds.

Dual-Driven Intrinsic Motivation. The RL agent is driven by the following intrinsic motivation:

$$r = r_{\text{env}} + \alpha_1 \cdot r_{\text{CAI}} + \alpha_2 \cdot r_{\text{ICM}}, \tag{10}$$

where α_1 and α_2 are the weighting coefficients for r_{CAI} and r_{ICM}, respectively, and r_{env} is the sparse reward of the environment.

Policy learning. For these goal-conditioned tasks, the policy receives the desired-goal vector g together with the task-relevant observation and selects actions according to $a \sim \pi_\theta(\cdot \mid s_R, g)$. The CCDM-derived partition filters only the observation dimensions in S_I; it does not remove the desired goal. All baselines receive the same desired-goal input, while CIM additionally excludes S_I from its observation input. This prevents the CIM policy from relying on nuisance variables and supports out-of-distribution generalization.

5 Experiments

We conduct experiments on multiple manipulation tasks under sparse-reward settings to answer the following questions: (1) In the base environment (without task-irrelevant state variables), how does our algorithm (CIM) perform compared to the baselines? (Sec. 5.2) (2) In environments with task-irrelevant state variables whose distribution varies periodically, how does CIM perform compared to the baselines? (Sec. 5.3) (3)Furthermore, when training is performed on a stable distribution and evaluation involves progressively more challenging distribution shifts, how does our method compare with existing baselines in terms of generalization ability? (Sec. 5.3) (4) What is the impact of key design choices, as assessed via ablations? (Sec. 5.4) All experimental results are obtained by running over 5 different random seeds. Implementation details, network architectures, training hyperparameters, and additional results are provided in Appendix D.

5.1 Experimental Setup

Environments. We evaluate our method on five sparse-reward manipulation tasks: Fetch/Push and Fetch/PickAndPlace from OpenAI Gym [48], a modified FetchPickAndPlaceRotTable (RotTable) environment with a rotatable table, and *pick-place* and *pick-place-wall* from Meta-World [49], which we adapt to sparse-reward settings. In these goal-conditioned tasks, the environment returns an observation o_t, an achieved goal g_t^{ach}, and a desired goal g_t^{des}. We denote the observation by $s_t = o_t$ and the desired goal by $g = g_t^{\text{des}}$; the achieved-goal information is already represented by the corresponding object-state dimensions in o_t. The CCDM models only o_t, whereas g is

Table 1: Policy performance under OOD generalization (success rate). OOD-1 and OOD-2 denote different levels of distribution shifts.

	GymFetch/Push		GymFetch/Pick		GymFetch/RotTable		Meta-world/*pick-place*		Meta-world/*pick-place-wall*	
Algorithm	OOD-1 (↑)	OOD-2 (↑)	OOD-1 (↑)	OOD-2 (↑)	OOD-1 (↑)	OOD-2 (↑)	OOD-1 (↑)	OOD-2 (↑)	OOD-1 (↑)	OOD-2 (↑)
CID	0.91±0.03	<u>0.45</u>±0.07	<u>0.80</u>±0.04	<u>0.32</u>±0.06	<u>0.57</u>±0.08	<u>0.26</u>±0.11	**0.92**±0.06	<u>0.54</u>±0.06	<u>0.11</u>±0.03	0.00±0.00
DuRND	<u>0.94</u>±0.02	0.24±0.03	0.64±0.05	0.13±0.02	0.26±0.06	0.15±0.06	0.80±0.02	0.33±0.09	0.08±0.04	0.00±0.00
Ensemble_Dis	0.66±0.04	0.21±0.05	0.79±0.13	0.28±0.09	0.01±0.01	0.00±0.00	0.86±0.07	0.39±0.05	0.00±0.00	0.00±0.00
HER	0.57±0.03	0.11±0.02	0.59±0.08	0.06±0.02	0.00±0.00	0.00±0.00	0.71±0.05	0.27±0.13	0.00±0.00	0.00±0.00
CIM (Ours)	**0.96**±0.04	**0.87**±0.01	**0.97**±0.01	**0.91**±0.05	**0.93**±0.06	**0.89**±0.07	<u>0.91</u>±0.04	**0.87**±0.07	**0.63**±0.10	**0.52**±0.09

provided separately to the policy and is excluded from the S_R/S_I partition. To test robustness and generalization, we further augment all of the above environments with task-irrelevant state variables, including immovable objects that do not influence the target object, object colors, and background noise. These modifications enable us to evaluate agent performance under simulated out-of-distribution settings while preserving the underlying causal mechanisms of the tasks.

Baselines. We compare CIM with four representative baselines: (a) HER [50], a standard goal-conditioned algorithm that addresses sparse rewards via experience relabeling; (b) Ensemble-Dis [51], which extends ICM by using an ensemble of forward dynamics models and drives exploration via intrinsic rewards based on prediction disagreement; (c) CID [6], which provides intrinsic rewards based on action influence estimated from a dense, non-factored dynamics model; and (d) DuRND [43], which proposes Dual Random Networks Distillation, a lightweight reward-shaping framework that uses separate positive and negative random network modules to jointly produce novelty and contribution rewards, thereby balancing directed exploration with goal-oriented exploitation in sparse-reward reinforcement learning. All algorithms are implemented based on Soft Actor-Critic (SAC) [52] combined with HER.

5.2 Base Performance

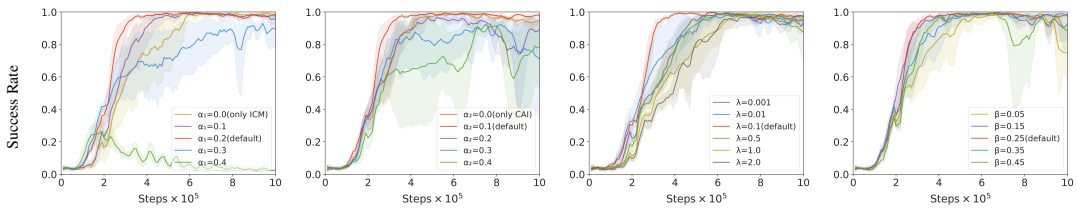
Figure 3a presents the training curves in the base sparse-reward environments without task-irrelevant state variables. It shows that our method achieves faster convergence, improved training stability, and a more favorable exploration-exploitation balance than the baselines. Notably, in the most challenging *pick-place-wall* environment, our method learns a successful policy reliably; by contrast, Ensemble-Dis and HER fail due to insufficient exploration, while CID and DRND succeed only sporadically, producing large variance across seeds.

We attribute these performance gains to two key contributing factors. First, we learn contextual causal dynamics by applying a causal-graph mask, ensuring that each predicted state dimension depends solely on its direct causal parents. This structural constraint yields a fine-grained dynamics model that enables the generation of precise intrinsic rewards. Second, we integrate two complementary signals: r_{ICM} to drive broad exploration early in training, and r_{CAI} to promote the acquisition of stable manipulation skills (e.g., pushing and grasping). Their complementarity yields superior empirical performance (see Sec. 5.4).

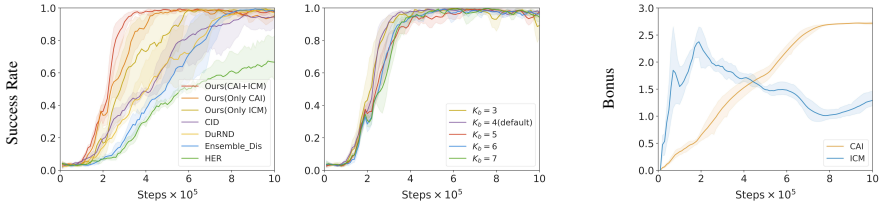
5.3 Training Robustness and Test Generalization

Training robustness. Figure 3b shows training performance in the presence of task-irrelevant state variables whose distribution is periodically shifted during training. Our method experiences only a limited degradation in convergence speed and stability and still reaches near-optimal performance within the training budget. By contrast, Ensemble-Dis and DRND, both of which rely solely on novelty-based intrinsic rewards, suffer pronounced performance drops after each distributional shift; CID, which primarily rewards skill acquisition, is less affected than Ensemble-Dis and DRND but nevertheless does not fundamentally remedy the problem and still performs poorly overall.

Policy generalization. Table 1 evaluates OOD generalization by training all models under a single stable distribution that includes task-irrelevant state variables and subsequently testing under distributional shifts of increasing difficulty. CIM experiences only a modest reduction in success rate under both OOD-1 and OOD-2 settings (See Appendix D.3). The remaining gap is largely attributable



(a) α_1. (b) α_2. (c) λ. (d) β.



(e) only r_{CAI}/r_{ICM}. (f) K_b. (g) Intrinsic bonus variation during training.

Figure 4: Ablation study.

to incomplete filtering of some task-irrelevant state dimensions. In contrast, all baselines experience a sharp decline in task success rates under distributional shifts, and in the difficult setting often fail entirely. These results demonstrate that CIM exhibits strong out-of-distribution generalization capability.

We further isolate the roles of contextual causal dynamics and state filtering in Sec. 5.5.

5.4 Ablation study

As shown in Fig. 4a, α_1 denotes the weight of the r_{CAI}: when α_1 is too small, learning becomes slow, whereas an excessively large α_2 can overwhelm the task reward and distract the agent from task completion, causing failures. As shown in Fig. 4b, α_2 denotes the weight of the r_{ICM}: removing r_{ICM} markedly slows early exploration, while an overly large α_2 encourages novelty-seeking interactions rather than goal achievement. As shown in Fig. 4c, λ is the L_1 regularization coefficient on the causal graph, which affects both downstream performance and causal-graph quality. An overly large λ suppresses important dimensions and can miss critical causal links, leading to degraded performance, whereas an overly small λ results in a near-dense model that diminishes the benefits of causal sparsity. Figure 4d shows β controls the strength of the commitment constraint, balancing representation fidelity and codebook stability. A small β leads to noisy code assignments, while a large β over-regularizes the encoder and harms flexibility and prediction.

Figure 4e compares variants using only r_{ICM}, only r_{CAI}, and the complete method against the baselines. The results show that, regardless of the reward formulation, introducing the contextual causal dynamics model alone already yields fine-grained intrinsic rewards and achieves better performance than the baselines. Figure 4f shows that our *Adaptive Codebook Fusion* mechanism makes the VQ-VAE architecture insensitive to the number of codebook entries. In our experiments, code vectors within the Euclidean-distance threshold of 2×10^{-3} are merged, yielding three causal subgraphs with negligible impact on downstream tasks. Figure 4g plots the temporal evolution of the r_{CAI} and r_{ICM} rewards during training. In early stages, r_{ICM} dominates to drive exploration and its average reward rises as interactions accumulate; as CCDM becomes more accurate, the r_{ICM} signal subsides and the r_{CAI} reward increases, stabilizing training and promoting robust manipulation skills rather than mere novelty-driven behavior.

5.5 Causal-Mechanism Ablation

We compare full CIM with two controls on GymFetch/PickAndPlace, training on Dist-0 and evaluating on OOD-2; the distribution definitions are given in Appendix D.3. The no-filter variant retains CCDM and the CAI+ICM reward design but gives the SAC policy the full state. The no-causal-

dynamics variant retains the reward design, replaces CCDM with a standard dense dynamics model, and uses a hand-designed mask to remove nuisance variables. Tables 2 and 3 report the training and OOD results.

Table 2: GymFetch/PickAndPlace training success rates on Dist-0 at the listed episode checkpoints (mean $\pm$ standard deviation).

Method / Episode	200	2,000	4,000	6,000	8,000	10,000	15,000	20,000
Ours w/o state filtering	0.035 ± 0.015	0.040 ± 0.016	0.175 ± 0.052	0.659 ± 0.202	0.880 ± 0.102	0.917 ± 0.083	0.965 ± 0.026	0.943 ± 0.031
Ours w/o causal dynamics	0.020 ± 0.017	0.052 ± 0.025	0.128 ± 0.039	0.472 ± 0.095	0.724 ± 0.075	0.845 ± 0.104	0.947 ± 0.044	0.936 ± 0.058
Ours (CIM)	0.048 ± 0.024	0.134 ± 0.021	0.438 ± 0.008	0.893 ± 0.087	0.940 ± 0.053	0.987 ± 0.007	0.977 ± 0.019	0.984 ± 0.010

Table 3: GymFetch/PickAndPlace success rates after training on Dist-0 and testing on OOD-2 (mean $\pm$ standard deviation).

Method	Train (Dist-0)	Test (OOD-2)
Ours w/o state filtering	0.943 ± 0.031	0.412 ± 0.265
Ours w/o causal dynamics	0.936 ± 0.058	0.918 ± 0.067
Ours (CIM)	0.984 ± 0.010	0.906 ± 0.082

The no-filter variant retains high training success (0.943 ± 0.031) but drops to 0.412 ± 0.265 on OOD-2, compared with 0.906 ± 0.082 for CIM. The dense-model variant with the hand-designed mask reaches 0.918 ± 0.067 on OOD-2, comparable to CIM within the reported variability. Thus, in this controlled setting, filtering is the main contributor to final OOD robustness; once a suitable mask is supplied, contextual causal dynamics does not further increase final OOD success. Its contribution is to discover the filtering structure automatically and improve training efficiency, as reflected in the earlier learning of full CIM.

6 Conclusions

This paper proposes CIM, a context-aware causal intrinsic motivation framework designed to improve robust exploration efficiency and out-of-distribution generalization in sparse-reward environments. By learning CCDM, CIM constructs intrinsic rewards based on sparse causal graphs and effectively filters task-irrelevant state variables at both the dynamics and policy levels. Extensive experiments across a range of robotic manipulation tasks demonstrate that CIM consistently outperforms existing baselines and maintains stable performance under distributional shifts.

Despite these promising results, the current framework still relies on accurate causal graph learning and is primarily evaluated on low-dimensional state-based observations. As a result, CIM cannot yet be directly applied to complex high-dimensional pixel-based environments, where reliable causal discovery becomes substantially more challenging due to partial observability and representation ambiguity. The future direction is to extend CIM to image-based settings by integrating representation learning techniques such as contrastive pretraining, masked reconstruction, and object-centric encoders with modular vision components. We aim to learn compact and sparse local causal structures to improve the scalability and generality of CIM.

Acknowledgments and Disclosure of Funding

This work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDB1010302, the Shanghai Municipal Science and Technology Project under Grant Nos. 25ZR1401370 and 25LN3200400, and the Shanghai Leading Talent Program of Eastern Talent Plan.

Competing interests. The authors declare no competing interests.

References

[1] Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. First return, then explore. *Nat.*, 590(7847):580–586, 2021. doi: 10.1038/S41586-020-03157-9. URL `https://doi.org/10.1038/s41586-020-03157-9`.

[2] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

[3] Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martín Arjovsky, Alexander Pritzel, Andew Bolt, et al. Never give up: Learning directed exploration strategies. *arXiv preprint arXiv:2002.06038*, 2020.

[4] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

[5] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018.

[6] Maximilian Seitzer, Bernhard Schölkopf, and Georg Martius. Causal influence detection for improving efficiency in reinforcement learning. *Advances in Neural Information Processing Systems*, 34:22905–22918, 2021.

[7] Artem Zholus, Yaroslav Ivchenkov, and Aleksandr Panov. Factorized world models for learning causal relationships. In *ICLR2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*, 2022.

[8] Zizhao Wang, Xuesu Xiao, Zifan Xu, Yuke Zhu, and Peter Stone. Causal dynamics learning for task-independent state abstraction. *arXiv preprint arXiv:2206.13452*, 2022.

[9] Zhongwei Yu, Jingqing Ruan, and Dengpeng Xing. Explainable reinforcement learning via a causal world model. *arXiv preprint arXiv:2305.02749*, 2023.

[10] Yudi Zhang, Yali Du, Biwei Huang, Ziyan Wang, Jun Wang, Meng Fang, and Mykola Pechenizkiy. Interpretable reward redistribution in reinforcement learning: A causal approach. *Advances in Neural Information Processing Systems*, 36:20208–20229, 2023.

[11] Muthukumar Pandaram, Jakob Hollenstein, David Drexel, Samuele Tosatto, Antonio Rodríguez-Sánchez, and Justus Piater. Dynamic sparsity: Challenging common sparsity assumptions for learning world models in robotic reinforcement learning benchmarks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 24727–24735, 2026.

[12] Inwoo Hwang, Yunhyeok Kwak, Suhyung Choi, Byoung-Tak Zhang, and Sanghack Lee. Fine-grained causal dynamics learning with quantization for improving robustness in reinforcement learning. *arXiv preprint arXiv:2406.03234*, 2024.

[13] Zhiyu Zhao, Haoxuan Li, Haifeng Zhang, Jun Wang, Francesco Faccio, Jürgen Schmidhuber, and Mengyue Yang. Curious causality-seeking agents learn meta causal world. *arXiv preprint arXiv:2506.23068*, 2025.

[14] Zhongwei Yu, Jingqing Ruan, and Dengpeng Xing. Learning causal dynamics models in object-oriented environments. *arXiv preprint arXiv:2405.12615*, 2024.

[15] Tianying Ji, Yongyuan Liang, Yan Zeng, Yu Luo, Guowei Xu, Jiawei Guo, Ruijie Zheng, Furong Huang, Fuchun Sun, and Huazhe Xu. Ace: Off-policy actor-critic with causality-aware entropy regularization. In *Forty-first International Conference on Machine Learning*, 2024.

[16] Xing Hu, Rui Zhang, Ke Tang, Jiaming Guo, Qi Yi, Ruizhi Chen, Zidong Du, Ling Li, Qi Guo, Yunji Chen, et al. Causality-driven hierarchical structure discovery for reinforcement learning. *Advances in Neural Information Processing Systems*, 35:20064–20076, 2022.

[17] Hongye Cao, Fan Feng, Tianpei Yang, Jing Huo, and Yang Gao. Causal information prioritization for efficient reinforcement learning. *arXiv preprint arXiv:2502.10097*, 2025.

[18] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

[19] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[20] Wenhao Ding, Haohong Lin, Bo Li, and Ding Zhao. Generalizing goal-conditioned reinforcement learning with variational causal reasoning. *Advances in Neural Information Processing Systems*, 35:26532–26548, 2022.

[21] Biwei Huang, Chaochao Lu, Liu Leqi, José Miguel Hernández-Lobato, Clark Glymour, Bernhard Schölkopf, and Kun Zhang. Action-sufficient state representation learning for control with structural constraints. In *International Conference on Machine Learning*, pages 9260–9279. PMLR, 2022.

[22] Zizhao Wang, Caroline Wang, Xuesu Xiao, Yuke Zhu, and Peter Stone. Building minimal and reusable causal state abstractions for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15778–15786, 2024.

[23] Fan Feng, Biwei Huang, Kun Zhang, and Sara Magliacane. Factored adaptation for non-stationary reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 31957–31971, 2022.

[24] Biwei Huang, Fan Feng, Chaochao Lu, Sara Magliacane, and Kun Zhang. Adarl: What, where, and how to adapt in transfer reinforcement learning. *arXiv preprint arXiv:2107.02729*, 2021.

[25] Silviu Pitis, Elliot Creager, and Animesh Garg. Counterfactual data augmentation using locally factored dynamics. *Advances in Neural Information Processing Systems*, 33:3976–3990, 2020.

[26] Núria Armengol Urpí, Marco Bagatella, Marin Vlastelica, and Georg Martius. Causal action influence aware counterfactual data augmentation. *arXiv preprint arXiv:2405.18917*, 2024.

[27] Pawel Ladosz, Lilian Weng, Minwoo Kim, and Hyondong Oh. Exploration in deep reinforcement learning: A survey. *Information Fusion*, 85:1–22, 2022.

[28] Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

[29] Leshem Choshen, Lior Fox, and Yonatan Loewenstein. Dora the explorer: Directed outreaching reinforcement action-selection. *arXiv preprint arXiv:1804.04012*, 2018.

[30] Rati Devidze, Goran Radanovic, Parameswaran Kamalaruban, and Adish Singla. Explicable reward design for reinforcement learning agents. *Advances in neural information processing systems*, 34:20118–20131, 2021.

[31] Haozhe Ma, Zhengding Luo, Thanh Vinh Vo, Kuankuan Sima, and Tze-Yun Leong. Highly efficient self-adaptive reward shaping for reinforcement learning. *arXiv preprint arXiv:2408.03029*, 2024.

[32] Zhengding Luo, Haozhe Ma, Dongyuan Shi, and Woon-Seng Gan. Gfanc-rl: Reinforcement learning-based generative fixed-filter active noise control. *Neural Networks*, 180:106687, 2024.

[33] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016.

[34] Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. Count-based exploration with neural density models. In *International conference on machine learning*, pages 2721–2730. PMLR, 2017.

[35] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

[36] Marlos C Machado, Marc G Bellemare, and Michael Bowling. Count-based exploration with the successor representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5125–5133, 2020.

[37] Sam Lobel, Akhil Bagaria, and George Konidaris. Flipping coins to estimate pseudocounts for exploration in reinforcement learning. In *International Conference on Machine Learning*, pages 22594–22613. PMLR, 2023.

[38] Aleksandr Ermolov and Nicu Sebe. Latent world models for intrinsically motivated exploration. *Advances in Neural Information Processing Systems*, 33:5565–5575, 2020.

[39] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

[40] Kai Yang, Jian Tao, Jiafei Lyu, and Xiu Li. Exploration and anti-exploration with distributional random network distillation. *arXiv preprint arXiv:2401.09750*, 2024.

[41] Tenglong Liu, Yang Li, Yixing Lan, Hao Gao, Wei Pan, and Xin Xu. Adaptive advantage-guided policy regularization for offline reinforcement learning. *arXiv preprint arXiv:2405.19909*, 2024.

[42] Hao Sun, Lei Han, Rui Yang, Xiaoteng Ma, Jian Guo, and Bolei Zhou. Exploit reward shifting in value-based deep-rl: Optimistic curiosity-based exploration and conservative exploitation via linear reward shaping. *Advances in neural information processing systems*, 35:37719–37734, 2022.

[43] Haozhe Ma, Fangling Li, Jing Yu Lim, Zhengding Luo, Thanh Vinh Vo, and Tze-Yun Leong. Catching two birds with one stone: Reward shaping with dual random networks for balancing exploration and exploitation. In *Forty-second International Conference on Machine Learning*, 2025.

[44] Judea Pearl. *Causality*. Cambridge university press, 2009.

[45] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[46] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[47] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

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

[49] 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 *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

[50] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

[51] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International conference on machine learning*, pages 5062–5071. PMLR, 2019.

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

A Theoretical Foundations for Contextual Causal Dynamics

In this section, we provide the theoretical underpinnings for the contextual Causal Dynamics Model (CCDM) utilized in our CIM framework. We first formulate the Factored Markov Decision Process (FMDP) as a Structural Causal Model (SCM), establish the necessary assumptions for causal discovery, and rigorously define Context-Specific Independence (CSI) and Local Causal Graphs (LCGs). Finally, we demonstrate how vector quantization theoretically achieves optimal state-action space decomposition for identifying these contextual causal structures.

A.1 Factored MDP as a Structural Causal Model

Building upon the standard MDP formulation introduced in Section 3.1, we analyze the environment's underlying mechanisms by explicitly considering its factored structure. Specifically, we cast the problem as a *Factored Markov Decision Process* (FMDP), where the continuous state and action spaces are composed of multiple distinct variables. Let $\mathbf{s}_t = [s_t^1, \dots, s_t^{d_\mathcal{S}}]^\top \in \mathcal{S}$ and $\mathbf{a}_t = [a_t^1, \dots, a_t^{d_\mathcal{A}}]^\top \in \mathcal{A}$ denote the state and action vectors at time step t, respectively.

This factored representation naturally allows us to model the transition dynamics $\mathcal{P}(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)$ as a dynamic Structural Causal Model (SCM). Following the SCM framework defined in Section 3.2, we define the endogenous variables for a single transition step as $\mathcal{V}_t = \mathbf{s}_t \cup \mathbf{a}_t \cup \mathbf{s}_{t+1}$. The causal mechanism generating each dimension of the future state, s_{t+1}^i, is governed by a structural assignment:

$$s_{t+1}^i := f_i\big(\mathrm{PA}(s_{t+1}^i; \mathcal{G}), u_t^i\big), \quad \forall i \in \{1, \dots, d_\mathcal{S}\}, \tag{11}$$

where $\mathrm{PA}(s_{t+1}^i; \mathcal{G}) \subseteq \mathbf{s}_t \cup \mathbf{a}_t$ denotes the set of direct causal parents in the global causal graph $\mathcal{G}$, and u_t^i represents the independent exogenous noise capturing environmental stochasticity.

A.2 Assumptions for Causal Discovery

To infer the underlying causal structure (i.e., the edges $\mathbf{a}_t \to \mathbf{s}_{t+1}$ and $\mathbf{s}_t \to \mathbf{s}_{t+1}$) from observational transition data $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1})$, we establish the following standard assumptions widely adopted in causal reinforcement learning:

Assumption 1 (Observational Sufficiency) *The set of observed variables is causally sufficient. There are no unobserved latent confounders that simultaneously exert causal influence on any variables in $(\mathbf{s}_t, \mathbf{a}_t)$ and the next state variables $\mathbf{s}_{t+1}$.*

Assumption 2 (Strict Temporal Ordering & Bipartite Structure) *The causal structure respects temporal constraints. Directed edges are permitted solely from variables at time t to variables at time $t+1$. Instantaneous edges (within the same time step t or $t+1$) or backward edges are strictly prohibited. Consequently, $\mathcal{G}$ is a directed bipartite graph.*

Assumption 3 (Global Markov Property) *The joint probability distribution $p(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1})$ satisfies the Global Markov Property with respect to $\mathcal{G}$. For any disjoint sets of variables $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$, d-separation in the graph implies conditional independence in the distribution: $\mathbf{X} \perp_\mathcal{G} \mathbf{Y} \mid \mathbf{Z} \implies \mathbf{X} \perp_p \mathbf{Y} \mid \mathbf{Z}$.*

Assumption 4 (Faithfulness Condition) *The distribution is faithful to the causal graph $\mathcal{G}$. This dictates that all conditional independence relations inherent in the distribution are structural entailments of the graph rather than spurious correlations arising from specific parameter cancellations: $\mathbf{X} \perp_p \mathbf{Y} \mid \mathbf{Z} \iff \mathbf{X} \perp_\mathcal{G} \mathbf{Y} \mid \mathbf{Z}$.*

Assumption 5 (Modular Independence) *The causal mechanism generating each component of the next state operates independently of the mechanisms generating other components. This aligns with the factored nature of the MDP, implying the transition probability factorizes as:*

$$\mathcal{P}(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t) = \prod_{i=1}^{d_\mathcal{S}} p\big(s_{t+1}^i \mid \mathrm{PA}(s_{t+1}^i; \mathcal{G})\big). \tag{12}$$

A.3 Identifiability of Global Causal Dynamics

Under Assumptions A.1 to A.5, the existence of specific causal links in the transition dynamics is strictly identifiable via conditional dependence analysis. Let $x \in \mathbf{s}_t \cup \mathbf{a}_t$ be any current state or action variable.

Proposition 1 (Identifiability of Global Edges) *A causal link exists from a current variable x to a next-state variable s^i_{t+1} (denoted $x \to s^i_{t+1}$) if and only if s^i_{t+1} remains conditionally dependent on x given all other current variables:*

$$x \to s^i_{t+1} \iff s^i_{t+1} \not\perp_p x \mid (\mathbf{s}_t \cup \mathbf{a}_t) \setminus \{x\}. \tag{13}$$

This proposition guarantees that by examining conditional mutual information, we can exactly recover the global causal graph $\mathcal{G}$.

A.4 Contextual Causal Relationships in FMDP

While the global causal graph $\mathcal{G}$ provides a macroscopic summary of dependencies, complex physical environments often exhibit heterogeneity. Specifically, certain causal mechanisms may only be activated under specific state-action subspaces.

A.4.1 Context-Specific Independence (CSI)

To capture valid causal relationships that hold only within restricted regions of the state-action space, we invoke the principle of Context-Specific Independence (CSI). Let $\mathcal{X} = \mathcal{S} \times \mathcal{A}$ define the joint domain. A subspace $\mathcal{D} \subseteq \mathcal{X}$ with positive measure is defined as a *context*.

Consider a target variable s^i_{t+1} and the set of all its potential global parents $\mathbf{V} = \mathrm{PA}(s^i_{t+1}; \mathcal{G})$. Under context $\mathcal{D}$, let $\mathbf{V}_{\mathrm{rel}} \subseteq \mathbf{V}$ denote the locally relevant predictors, and $\mathbf{V}_{\mathrm{irr}} = \mathbf{V} \setminus \mathbf{V}_{\mathrm{rel}}$ denote the locally irrelevant ones. The variable s^i_{t+1} is contextually independent of $\mathbf{V}_{\mathrm{irr}}$ given $\mathbf{V}_{\mathrm{rel}}$ within context $\mathcal{D}$ if:

$$p\big(s^i_{t+1} \mid \mathbf{v}_{\mathrm{rel}}, \mathbf{v}_{\mathrm{irr}}\big) = p\big(s^i_{t+1} \mid \mathbf{v}_{\mathrm{rel}}\big), \quad \forall (\mathbf{s}_t, \mathbf{a}_t) \in \mathcal{D}. \tag{14}$$

A.4.2 Local Causal Graph (LCG)

Building upon CSI, as introduced in Section 3.3, we formally define the *Local Causal Graph* (LCG) $\mathcal{G}_{\mathcal{D}} = (\mathcal{V}_{\mathcal{D}}, \mathcal{E}_{\mathcal{D}})$ for context $\mathcal{D}$. The edge set $\mathcal{E}_{\mathcal{D}}$ includes a directed edge $x \to s^i_{t+1}$ if and only if x belongs to the minimal set of locally relevant parents, denoted as $\mathrm{PA}(s^i_{t+1}; \mathcal{G}_{\mathcal{D}})$.

The global causal graph is effectively the union of all such local dependencies: $\mathcal{G} = \bigcup_{\mathcal{D}} \mathcal{G}_{\mathcal{D}}$. Consequently, an LCG $\mathcal{G}_{\mathcal{D}}$ exhibits significantly higher sparsity than the global graph $\mathcal{G}$, accurately reflecting the physical reality that many interactions are localized events (e.g., an agent only affects an object's state when making physical contact).

A.5 State-Action Space Decomposition via Vector Quantization

Identifying these granular structures requires partitioning the continuous state-action space without prior knowledge of the contexts. In our CCDM module, this is achieved by decomposing $\mathcal{X}$ into K disjoint subspaces $\{\mathcal{D}_k\}_{k=1}^K$. We utilize a Vector Quantization (VQ-VAE) bottleneck with a codebook $\mathcal{C} = \{e_1, \ldots, e_K\}$ to learn this assignment.

The mapping of any sample $(\mathbf{s}_t, \mathbf{a}_t)$ to a specific context k is determined by nearest-neighbor lookup in the latent space:

$$k = \arg \min_{j \in \{1,\ldots,K\}} \big\| \mathrm{enc}(\mathbf{s}_t, \mathbf{a}_t) - e_j \big\|_2. \tag{15}$$

The transition dynamics are thus modeled as a mixture of context-aware experts:

$$\hat{\mathcal{P}}(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t) = \sum_{k=1}^{K} \mathbb{I}\big[(\mathbf{s}_t, \mathbf{a}_t) \in \mathcal{D}_k\big] \prod_{i=1}^{d_{\mathcal{S}}} \hat{p}\big(s^i_{t+1} \mid \mathrm{PA}(s^i_{t+1}; \mathcal{G}_k)\big). \tag{16}$$

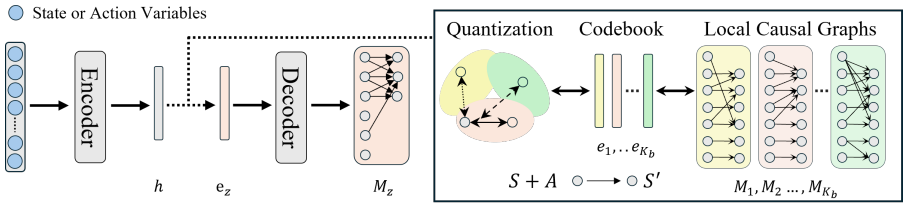


Figure 5: VQ-VAE.

To simultaneously optimize the space decomposition (the VQ codebook) and identify the corresponding LCGs, CCDM implicitly optimizes a regularized maximum likelihood score $\mathcal{J}$:

$$\mathcal{J}\left(\{\mathcal{G}_k, \mathcal{D}_k\}_{k=1}^{K}\right) = \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t)}\left[\sum_{k=1}^{K} \mathbb{I}\left[(\mathbf{s}_t, \mathbf{a}_t) \in \mathcal{D}_k\right] \sum_{i=1}^{d_{\mathcal{S}}} \left(\log \hat{p}\left(s_{t+1}^i \mid \mathrm{PA}(s_{t+1}^i; \mathcal{G}_k)\right) - \lambda |\mathcal{G}_k|\right)\right], \tag{17}$$

where $|\mathcal{G}_k|$ denotes the edge count (sparsity level) of the k-th local graph, and λ is a sparsity penalty. Theoretical findings in causal discovery suggest that maximizing this score guarantees the identifiability of the true contexts and their associated sparse causal mechanisms.

A.6 Gumbel-Sigmoid

To enable gradient-based optimization over the discrete space of graph structures, the binary adjacency matrix is approximated via a continuous relaxation using the Gumbel-Sigmoid reparameterization. Distinct from the Gumbel-Softmax trick, which enforces categorical exclusivity (suitable for selecting a single element), the Gumbel-Sigmoid models the existence of each directed edge as an independent Bernoulli random variable. This formulation permits nodes to have multiple concurrent parents and facilitates the differentiable identification of complex dependency structures within the local causal graphs through backpropagation.

A.7 Vector Quantization (VQ-VAE)

Figure 5 illustrates the detailed VQ-VAE architecture used in our framework. The encoder maps the input transition pair (s, a) to a continuous latent embedding h. A vector quantization module then assigns this embedding to the nearest entry e_z in a learned codebook $\mathcal{C}$, where each codebook entry corresponds to a specific local causal graph. Subsequently, a decoder reconstructs e_z to produce edge logits. These logits are processed via Gumbel-Sigmoid sampling and reshaped into a causal adjacency matrix of size $(d_{\mathcal{S}} + d_{\mathcal{A}}) \times d_{\mathcal{S}}$, explicitly defining the active causal dependencies for prediction.

Codebook collapsing. Training discrete latent representations via Vector Quantization (VQ) is prone to codebook collapse, a phenomenon where the encoder maps the majority of input samples to a small subset of codebook vectors, leaving the remaining codes underutilized. In our framework, such collapse is detrimental, as it prevents the model from distinguishing fine-grained causal contexts, potentially degenerating the learned dynamics into a single global model. To mitigate this, we employ the Exponential Moving Average (EMA) [45]strategy to update the codebook embeddings. Instead of optimizing the codebook via direct gradient descent, EMA updates each code vector based on the moving average of the encoder outputs assigned to it, which has been shown to significantly enhance utilization rates and training stability.

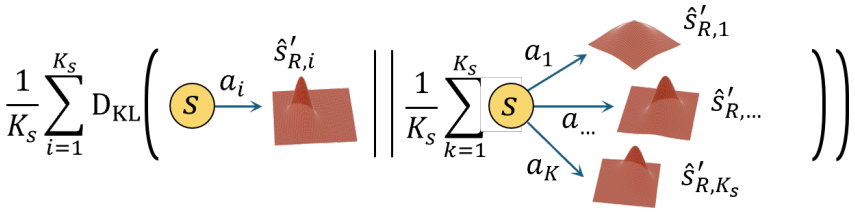


Figure 6: $C^R(s, g)$.

B Graph-Based Conditional Mutual Information

B.1 Formula Derivation

The causal action influence on the relevant state variables S'_R is quantified by the conditional mutual information $C^R(s)$. The derivation is as follows:

$$\begin{aligned} C^R(s) &= I(S'_R; A \mid S = s) \\ &= D_{\mathrm{KL}}\Big(P_{S'_R,A|s} \;\Big\|\; P_{S'_R|s} \otimes \mathcal{U}(\mathcal{A})\Big) \\ &= \mathbb{E}_{S'_R,A|s}\left[\log \frac{p(s'_R, a \mid s; M_z)}{p(s'_R \mid s; M_z)\, u_{\mathcal{A}}(a)}\right] \\ &= \mathbb{E}_{S'_R,A|s}\left[\log \frac{p(s'_R \mid s, a; M_z)\, u_{\mathcal{A}}(a)}{p(s'_R \mid s; M_z)\, u_{\mathcal{A}}(a)}\right] \\ &= \mathbb{E}_{S'_R,A|s}\left[\log \frac{p(s'_R \mid s, a; M_z)}{\int p(s'_R \mid s, a'; M_z)\, u_{\mathcal{A}}(a')\, da'}\right] \\ &= \mathbb{E}_{A\sim\mathcal{U}(\mathcal{A})}\left[D_{\mathrm{KL}}\Big(P_{S'_R|s,a;M_z} \;\Big\|\; P_{S'_R|s;M_{1,\ldots,K_b}}\Big)\right]. \end{aligned} \tag{18}$$

B.2 Monte-Carlo Estimation of Causal Action Influence

In practice, the exact computation of the causal action influence reward r_{CAI}, which corresponds to the mutual information $C^R(s)$, is intractable due to the integral over the continuous action space required to compute the marginal distribution $p(s'_R \mid s; M_{1,\ldots,K_b})$. To address this issue, we employ a finite-sample approximation based on a variational lower bound of the mutual information.

We first utilize the parameterized contextual causal dynamics model $p_\theta(s'_R \mid s, a; M_z)$ learned from observational data. We then approximate both the outer expectation over actions and the intractable marginal density $p(s'_R \mid s; M_{1,\ldots,K_b})$ using Monte-Carlo sampling. Let $\{a^{(1)}, \ldots, a^{(K_s)}\}$ be K_s i.i.d. samples from the fixed uniform distribution $\mathcal{U}(\mathcal{A})$. The marginal distribution is approximated as a finite mixture of the conditional predictions:

$$p(s'_R \mid s; M_{1,\ldots,K_b}) \approx \frac{1}{K_s}\sum_{k=1}^{K_s} p_\theta(s'_R \mid s, a^{(k)}; M_z). \tag{19}$$

Substituting this approximation into the mutual information objective yields the following estimator, which is formally defined below and illustrated in Fig. 6:

$$\hat{C}^R(s) = \frac{1}{K_s}\sum_{i=1}^{K_s} D_{\mathrm{KL}}\left(p_\theta(s'_R \mid s, a^{(i)}; M_z) \;\Big\|\; \frac{1}{K_s}\sum_{k=1}^{K_s} p_\theta(s'_R \mid s, a^{(k)}; M_z)\right). \tag{20}$$

This estimator constitutes a consistent lower bound on the true conditional mutual information $I(S'_R; A \mid S = s)$, which tightens as the number of Monte-Carlo samples K_s increases.

Table 4: CAI and interaction states along a successful grasping trajectory.

Step	1	2	3	4	5	6	7	8	9	10
CAI	0.125	0.161	0.719	1.424	2.323	2.764	3.227	3.190	3.225	3.208
EE-to-object distance (m)	0.147	0.098	0.062	0.019	0.005	0.005	0.005	0.005	0.005	0.005
Contact / Grasp	✗/✗	✗/✗	✗/✗	✗/✗	✓/✗	✓/✓	✓/✓	✓/✓	✓/✓	✓/✓

Table 5: Success rates at training checkpoints for different action-sample counts K_s (mean $\pm$ standard deviation).

K_s / Episode	200	1,000	2,000	4,000	5,000	6,000	7,000	8,000	10,000	15,000	20,000
8	0.031 ± 0.017	0.030 ± 0.008	0.046 ± 0.020	0.154 ± 0.085	0.498 ± 0.060	0.750 ± 0.165	0.894 ± 0.087	0.952 ± 0.044	0.971 ± 0.026	0.988 ± 0.008	0.911 ± 0.087
16	0.034 ± 0.014	0.019 ± 0.008	0.054 ± 0.019	0.282 ± 0.122	0.662 ± 0.067	0.847 ± 0.133	0.909 ± 0.081	0.923 ± 0.066	0.984 ± 0.015	0.991 ± 0.005	0.951 ± 0.024
32	0.032 ± 0.011	0.032 ± 0.016	0.050 ± 0.031	0.333 ± 0.155	0.715 ± 0.166	0.893 ± 0.081	0.947 ± 0.027	0.985 ± 0.009	0.985 ± 0.007	0.980 ± 0.011	0.963 ± 0.014
64	0.049 ± 0.010	0.033 ± 0.011	0.061 ± 0.023	0.203 ± 0.122	0.604 ± 0.091	0.795 ± 0.113	0.894 ± 0.075	0.926 ± 0.043	0.952 ± 0.033	0.986 ± 0.012	0.975 ± 0.005

B.3 Action-Sampling Ablation and CAI Analysis

We examine how CAI reflects controllable interactions and how the number of sampled actions K_s affects policy learning and estimation stability. Unless noted otherwise, candidate actions are sampled i.i.d. from the fixed uniform distribution $\mathcal{U}(\mathcal{A})$ defined in Eq. 7.

CAI along a grasping trajectory. Table 4 reports CAI and interaction states for a successful grasping trajectory. CAI rises from 0.125–1.424 before contact to 2.323 at first contact, and remains around 3.2 during grasping. This illustrates that CAI reflects action-conditioned control effects rather than merely policy randomness. General novelty signals such as Ensemble-Dis do not directly distinguish these levels of controllability and may diminish as their dynamics models converge, whereas CAI remains high for effective contact and grasping actions. In this sparse-reward setting, SAC's entropy regularization alone is insufficient to learn pick-and-place.

Effect of K_s on policy learning. We compare success rates across $K_s \in \{8, 16, 32, 64\}$ at the same training checkpoints. All four settings learn successful policies; $K_s = 32$ shows the strongest early-to-middle learning (4,000–10,000 episodes), while final performance is high across settings.

Stability of CAI estimates. We evaluate the converged policies over 100 episodes initialized from grasping states and summarize the CAI estimates for each K_s in Table 6.

By Eq. 20, the finite-sample CAI satisfies $0 \le \hat{C}^R(s) \le \log K_s$. Although the upper bound grows with K_s, the normalized CAI remains between 0.859 and 0.900, and the standard-deviation-to-mean ratios remain between 0.0381 and 0.0617. These results indicate stable estimates across the tested sample counts and support $K_s = 32$ as a practical choice with strong early policy learning.

C Algorithm

C.1 Adaptive Codebook Fusion

After each CCDM optimization and its codebook EMA update, we check codebook-vector pairs using the Euclidean distance $d_z(i, j) = \|e_i - e_j\|_2$. A pair is merged when $d_z(i, j) < 2 \times 10^{-3}$. For a pair (e_i, e_j) selected for merging, we update the retained vector as

$$e_i^{\text{new}} = (1 - r)e_i + re_j, \qquad r = \frac{N(j)}{N(i) + N(j)}, \tag{21}$$

where $N(i)$ and $N(j)$ are the numbers of assignments to the corresponding entries during the forward pass. We then reinitialize the freed vector e_j from a normal distribution and reset its usage count to zero; it remains available for subsequent nearest-neighbor assignments.

C.2 CIM Training Procedure

The complete algorithm is presented in Algo. 1. Initialize the relevance mask to all ones ($S_R = S$) and keep the full observation for the first 1,000 episodes while CCDM warms up. Thereafter, recompute

Table 6: CAI estimates across action-sample counts, evaluated from grasping states.

Metric	$K_s = 8$	$K_s = 16$	$K_s = 32$	$K_s = 64$
CAI score	1.786 $\pm$ 0.068	2.447 $\pm$ 0.111	3.100 $\pm$ 0.159	3.742 $\pm$ 0.231
$\log K_s$	2.079	2.773	3.466	4.159
CAI / $\log K_s$	0.859	0.883	0.895	0.900
Std. / mean	0.0381	0.0454	0.0513	0.0617

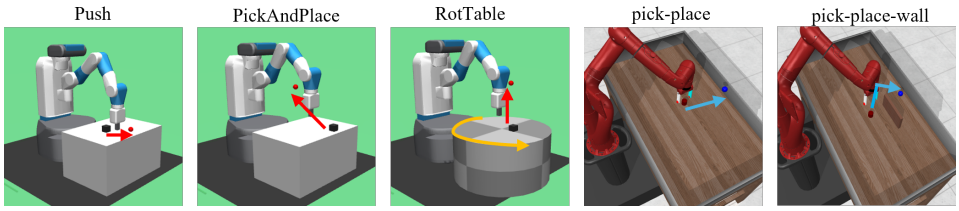


Figure 7: Manipulation environments.

the mask after each episode by aggregating action-parent edges across all LCGs. Applying the mask elementwise suppresses filtered features without changing the policy input dimension; aggregation also prevents one erroneous local graph from permanently removing a useful feature.

Algorithm 1 contextual **C**ausal dynamics-based **I**ntrinsic **M**otivation (CIM)

Input: Observation S, Desired goal G, Action A, Hyperparameters λ (Sparsity), α_1 (r_{CAI}), α_2 (r_{ICM}), β, K_b
Output: Optimized Policy π_θ, CCDM ϕ, LCGs $M_{1,\dots,K_b}$
Initialize: CCDM parameters $\phi = \{\text{Encoder } \phi_{\text{enc}}, \text{Decoder } \phi_{\text{dec}}, \text{Dynamics } \phi_{\text{dyn}}\}$; Policy agent π_θ; mask $\mathbf{m}_R = \mathbf{1}$ ($S_R = S$); Codebook C_{vq}
for episode $k = 1$ to K **do**
  Receive initial observation s_0 and desired goal g
  for step $t = 0$ to $T - 1$ **do**
    // 1. Interaction
    Select action $a_t \sim \pi_\theta(\cdot \mid \mathbf{m}_R \odot s_t, g)$
    Execute a_t, observe s_{t+1} and $r_{\text{env},t}$
    Store transition $(s_t, a_t, s_{t+1}, r_{\text{env},t}, g)$ in $\mathcal{D}$
    // 2. Training Phase
    Sample batch $B = \{(s, a, r_{\text{env}}, s', \tilde{g})\}$ from $\mathcal{D}$, with goals relabeled by HER
    Infer latent $h = \phi_{\text{enc}}(s, a)$, quantize to e_z of C_{vq}, then get $M_z = \phi_{\text{dec}}(e_z)$ (Eq. 3)
    Predict $\hat{s}'_R = \phi_{\text{dyn}}(s, a; M_z)$ (Eq. 4)
    Update ϕ by minimizing $\mathcal{L}_{CCDM}$ (Eq. 9) and update C_{vq} via EMA
    Apply adaptive codebook fusion to pairs satisfying the distance threshold (Appendix C.1)
    Compute influence reward r_{CAI} (Eq. 7) and r_{ICM} (Eq. 8)
    Total reward r_{total} (Eq. 10)
    Update policy agent π_θ using r_{total} conditioned on $(\mathbf{m}_R \odot s, \tilde{g})$
  end for
  if $k \geq 1000$ **then**
    Decode all LCGs $M_1, \dots, M_{K_b} = \phi_{\text{dec}}(C_{vq})$ and compute I using Eq. 5
    Update $\mathbf{m}_R = (\mathbb{I}[i \notin I])_{i=1}^{d_S}$
  end if
end for

Table 7: Comparison of CIM with baselines.

	HER	Ensemble-Dis	CID	DuRND	CIM(Ours)
Learns dynamics	✗	✓	✓	✗	✓
Out-of-distribution	✗	✗	✗	✗	✓
Balances exploration & exploitation	✗	✗	✗	✓	✓

D Experimental Details

D.1 Baselines

Table 7 summarizes the key differences between our method and the baselines. Compared to existing approaches, our method not only provides more precise reward signals through causal modeling, enabling efficient exploration, but also demonstrates robustness to environmental perturbations and better generalization under out-of-distribution settings.

Hindsight Experience Replay (HER) [50]: HER is an experience replay strategy for sparse and binary reward settings in reinforcement learning. It addresses the scarcity of positive signals by relabeling failed trajectories with goals that were actually achieved later in the same episode (e.g., the final achieved state). This transforms unsuccessful rollouts into valid training data, enabling agents to learn from both success and failure. By training goal-conditioned policies on these relabeled transitions, HER significantly improves sample efficiency for off-policy algorithms such as SAC without requiring explicit reward shaping.

Ensemble Disagreement (Ensemble-Dis) [51]: Ensemble-Dis is an intrinsic motivation method designed to improve exploration in stochastic and sparse-reward environments. It trains an ensemble of forward dynamics models that predict the next-state embeddings from the current state and action. Exploration rewards are computed based on the disagreement (variance) among ensemble predictions.

Causal Influence Detection (CID) [6]: CID enhances exploration by explicitly measuring the action influence of an agent's actions on the environment. It quantifies a local causal action influence score, typically modeled as conditional mutual information between actions and future states, estimated using a learned dynamics model.

Dual Random Networks Distillation (DuRND) [43]: DuRND is a reward-shaping framework that balances exploration and exploitation in sparse-reward settings using two complementary random network modules. A positive module is trained on trajectories that achieve task success, while a negative module is trained on all other trajectories. Intrinsic rewards are derived from both prediction errors: a novelty signal encouraging exploration of unfamiliar states, and a contribution signal estimating a state's likelihood of leading to success. This dual mechanism enables more stable exploration early in training and more goal-directed behavior later, while remaining computationally efficient.

D.2 Base Environments

We conduct experiments on five continuous-control robotic manipulation tasks, as illustrated in Fig. 7. To evaluate exploration and skill acquisition, all environments are configured under a sparse-reward setting: the agent receives an environment reward of 0 upon successful task completion and -1 otherwise. The task specifications are as follows:

- **GymFetch/Push** ($T = 50$): The agent controls a robotic arm to push a object to a designated target location on the table. The state space has 25 dimensions, the action space has 4 dimensions, and the goal space has 3 dimensions. Each episode lasts for 50 environment steps.
- **GymFetch/PickAndPlace** ($T = 50$): The agent must grasp a object and lift it to a target coordinate in the air or on the table. This task uses the same dimensionality as Push, with a state space of 25 dimensions, an action space of 4 dimensions, and a goal space of 3 dimensions. Each episode lasts for 50 environment steps.
- **GymFetch/RotTable** ($T = 100$): This is a dynamic variant featuring a periodically rotating table. As a result, when the target is placed on the tabletop, it remains in constant motion. This prevents

the agent from succeeding through simple pushing and requires it to acquire stable and secure grasping skills.To account for the table kinematics, the state space is expanded to 29 dimensions by incorporating the rotation period and angular velocity of the table, while the action space has 4 dimensions and the goal space has 3 dimensions. Each episode lasts for 100 environment steps.

- **Meta-World/*pick–place*** ($T = 250$): This is a sparse-reward adaptation of the standard Meta-World *pick-place* task. The state space has 39 dimensions, the action space has 4 dimensions, and the goal space has 3 dimensions. Each episode lasts for 250 environment steps.
- **Meta-World/*pick-place-wall*** ($T = 250$): This is a significantly more challenging sparse-reward setting in which a wall blocks the direct path to the target. The agent must grasp the object and navigate around the wall before placing it at the destination. The state space has 39 dimensions, the action space has 4 dimensions, and the goal space has 3 dimensions. Each episode lasts for 250 environment steps.

D.3 Augmented Environments with Task-Irrelevant Variables

To evaluate the robustness and out-of-distribution (OOD) generalization capabilities of our method, we follow the experimental protocol introduced in Causal Dynamics Learning (CDL) [8] by augmenting the original state spaces with $N = 12$ task-irrelevant variables. These variables exert absolutely no causal influence on the underlying transition dynamics or the reward function. The 12 extra dimensions consist of:

1. The 3D position of the immovable object, $\mathbf{p}_{\text{irr}} = (x, y, z) \in \mathbb{R}^3$.
2. The RGB color channels of the immovable object, $\mathbf{c}_{\text{irr}} \in \mathbb{R}^3$.
3. The RGB color channels of the target object, $\mathbf{c}_{\text{target}} \in \mathbb{R}^3$.
4. Three dimensions of background random noise, $\mathbf{n}_{\text{noise}} \in \mathbb{R}^3$.

Crucially, the values of these variables are generated via random uniform sampling within fixed intervals. Distributional shifts are induced strictly by altering these sampling intervals.

Robustness Training Setup: During the training phase, the agent is trained on a base distribution (**Dist-0**), where the variables are sampled as follows: $\mathbf{p}_{\text{irr}} \sim \mathcal{U}([0,3] \times [0,3] \times [0,1])$; $\mathbf{c}_{\text{irr}} \sim \mathcal{U}([0,10]^3)$; $\mathbf{c}_{\text{target}} \sim \mathcal{U}([0,10]^3)$; and $\mathbf{n}_{\text{noise}} \sim \mathcal{U}([-5,5]^3)$.

Three additional distribution variants (**Dist-1**, **Dist-2**, and **Dist-3**) are defined to induce progressively stronger distribution shifts:

Dist-1: $\mathbf{p}_{\text{irr}} \sim \mathcal{U}([5,10] \times [5,10] \times [1,2])$; $\mathbf{c}_{\text{irr}} \sim \mathcal{U}([10,20]^3)$; $\mathbf{c}_{\text{target}} \sim \mathcal{U}([10,20]^3)$; $\mathbf{n}_{\text{noise}} \sim \mathcal{U}([-10,10]^3)$.

Dist-2: $\mathbf{p}_{\text{irr}} \sim \mathcal{U}([10,20] \times [10,20] \times [2,5])$; $\mathbf{c}_{\text{irr}} \sim \mathcal{U}([20,50]^3)$; $\mathbf{c}_{\text{target}} \sim \mathcal{U}([20,50]^3)$; $\mathbf{n}_{\text{noise}} \sim \mathcal{U}([-20,20]^3)$.

Dist-3: $\mathbf{p}_{\text{irr}} \sim \mathcal{U}([20,30] \times [20,30] \times [5,10])$; $\mathbf{c}_{\text{irr}} \sim \mathcal{U}([50,100]^3)$; $\mathbf{c}_{\text{target}} \sim \mathcal{U}([50,100]^3)$; $\mathbf{n}_{\text{noise}} \sim \mathcal{U}([-50,50]^3)$.

Policy Generalization Testing Setup: After training the policy on the stable **Dist-0** distribution, we conduct out-of-distribution (OOD) evaluations under two specific scenarios:

OOD-1: An expanded version of Dist-0: $\mathbf{p}_{\text{irr}} \sim \mathcal{U}([0,10] \times [0,10] \times [0,2])$; $\mathbf{c}_{\text{irr}} \sim \mathcal{U}([0,20]^3)$; $\mathbf{c}_{\text{target}} \sim \mathcal{U}([0,20]^3)$; and $\mathbf{n}_{\text{noise}} \sim \mathcal{U}([-10,10]^3)$.

OOD-2: A severe distribution shift: $\mathbf{p}_{\text{irr}} \sim \mathcal{U}([10^3,10^4] \times [10^3,10^4] \times [10^2,10^3])$; $\mathbf{c}_{\text{irr}} \sim \mathcal{U}([155,255]^3)$; $\mathbf{c}_{\text{target}} \sim \mathcal{U}([155,255]^3)$; and $\mathbf{n}_{\text{noise}}$ is sampled from a disjoint extreme-range distribution, i.e., $\mathbf{n}_{\text{noise}} \sim \mathcal{U}\big(([-10^4,-10^2] \cup [10^2,10^4])^3\big)$.

D.4 Additional Results

Reward-sparsity verification for HER/SAC. To test whether HER/SAC's *pick-place-wall* failure stems from reward sparsity, we compare dense and sparse Meta-World rewards with identical SAC settings. The original dense reward provides stage-wise guidance for approach, grasping, lifting,

Table 8: HER/SAC success rates under dense and sparse rewards (mean ± standard deviation).

Reward / Episode	100	2,000	4,000	6,000	8,000	10,000	13,000	16,000	20,000
Dense	0.00 ± 0.00	0.00 ± 0.00	0.21 ± 0.10	0.50 ± 0.17	0.54 ± 0.29	0.72 ± 0.19	0.77 ± 0.05	0.84 ± 0.04	0.88 ± 0.06
Sparse	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00

Table 9: Success rates under training and broken-correlation test distributions (mean ± standard deviation).

Method	Train	Test
CIM (Ours)	0.989 ± 0.005	0.879 ± 0.037
Ensemble-Dis	0.896 ± 0.084	0.286 ± 0.217
CID	0.914 ± 0.116	0.294 ± 0.186
DuRND	0.875 ± 0.188	0.155 ± 0.108
HER	0.776 ± 0.133	0.093 ± 0.030

and transport, including grasp-maintenance and intermediate-position bonuses and a 10 bonus when object–target distance is below 0.05 [49]. Our only change is the reward (-1 per step, 0 on success); the success threshold, horizon, termination, and action/observation normalization are unchanged. HER uses future relabeling with $p = 0.8$.

HER/SAC learns under the dense reward, consistent with Fig. 11 of Meta-World [49], but never succeeds under the sparse reward. This confirms that its failure is primarily due to reward sparsity rather than inadequate tuning.

OOD robustness to broken structural correlations. To test robustness when nuisance variables are correlated with task state, we add an immovable object m to GymFetch/Push. During training, m is placed $10\,\mathrm{cm}$ to the left of the target object's initial position, and its color (red, blue, green, or purple) is determined by which of the four regions formed by the table's x- and y-centerlines contains the target. At test time, we break both correlations: m is placed $15\,\mathrm{cm}$ behind the target's initial position, and its RGB vector is sampled independently as $\mathbf{c}_m \sim \mathcal{U}([0, 100]^3)$. The object remains immovable and cannot be changed by actions. Table 9 reports success rates on the training and test distributions.

All methods achieve some success while the training correlations remain intact, but breaking these correlations at test time sharply reduces baseline performance. CIM retains substantially stronger test performance. Because m's position and color are invariant to actions, CCDM can treat them as nuisance dimensions rather than action-relevant causes, reducing reliance on the training-time heuristics. This result extends CIM's robustness beyond independent noise to shifts in structural correlations.

Controllable Noisy-TV experiment and CAI ablation. We evaluate a state-based Noisy-TV variant in GymFetch, since the image-based benchmark [5] is not included in the released code and is outside our state-based setup. A reachable, fixed TV region contains an immovable object whose RGB state is the display. When the end effector enters the region and selects the channel-switch action, a channel $i \in \{0, \ldots, 9\}$ is sampled uniformly and each RGB component independently from $\mathcal{U}([10i, 10(i + 1)))$. The agent can trigger a display change but cannot predict its outcome. We compare full CIM, Ensemble-Dis, and CIM with CAI ablated ($\alpha_1 = 0$; all other settings unchanged). CAI candidate actions come from a fixed uniform distribution, independent of the learned policy. Table 10 gives success rates (mean ± standard deviation) at the listed episode checkpoints.

Table 10: Noisy-TV task success rates at training checkpoints (mean ± standard deviation).

Method / Episode	200	1,000	2,000	4,000	5,000	6,000	7,000	8,000	10,000	15,000	20,000
CIM	0.038 ± 0.016	0.029 ± 0.011	0.015 ± 0.013	0.175 ± 0.050	0.310 ± 0.059	0.445 ± 0.015	0.525 ± 0.025	0.683 ± 0.091	0.928 ± 0.030	0.969 ± 0.029	0.936 ± 0.047
Ensemble-Dis	0.025 ± 0.005	0.034 ± 0.013	0.031 ± 0.012	0.056 ± 0.021	0.085 ± 0.040	0.124 ± 0.073	0.283 ± 0.161	0.388 ± 0.265	0.609 ± 0.335	0.624 ± 0.315	0.651 ± 0.346
CIM ($\alpha_1 = 0$, ICM only)	0.046 ± 0.012	0.033 ± 0.012	0.020 ± 0.015	0.052 ± 0.016	0.159 ± 0.121	0.224 ± 0.169	0.313 ± 0.129	0.358 ± 0.167	0.475 ± 0.210	0.594 ± 0.198	0.684 ± 0.253

The ablation shows that removing CAI slows learning and increases variability, although ICM-only CIM can still succeed. Thus, CAI contributes to reliable escape from the TV distraction but is not the

sole source of task success. The learned TV-interaction graph contains an edge from the channel-switch action to the TV RGB state; its context accounts for 0.094 ± 0.039 of dynamics-prediction assignments.

Under the same fixed uniform candidate-action distribution, CAI is approximately 0.400 in the TV region and 3.100 during object contact and grasping. This contrast provides a stronger intrinsic signal for task-relevant interaction, although the stochastic TV still slows learning.

We further inspect the first 12 steps of a successful 50-step episode. The end effector starts inside the Noisy-TV region, while the object and goal are outside it. Table 11 shows that the agent leaves the TV region by Step 4, approaches the object, establishes contact at Step 9, and grasps it at Step 10. CAI rises as the episode progresses toward task-relevant interaction.

Table 11: CAI and interaction states during the first 12 steps of one successful Noisy-TV episode.

Metric / Step	1	2	3	4	5	6	7	8	9	10	11	12
CAI	0.429	0.379	0.391	0.095	0.113	0.612	1.503	2.103	2.843	3.078	3.093	3.149
EE-to-object distance (m)	0.416	0.384	0.358	0.322	0.275	0.123	0.027	0.014	0.005	0.005	0.005	0.005
Contact / Grasp	✗/✗	✗/✗	✗/✗	✗/✗	✗/✗	✗/✗	✗/✗	✗/✗	✓/✗	✓/✓	✓/✓	✓/✓
Inside Noisy-TV	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗

Final performance. Consistent with the results in Fig. 3a, Table 12 (which summarizes the final performance over 100 test episodes in terms of mean $\pm$ standard deviation) further demonstrates that CIM achieves the highest episodic returns across all evaluated environments. Notably, in these tasks, higher returns are directly correlated with more efficient task completion, as successful and faster rollouts accumulate less penalty over time. Therefore, the superior return of CIM not only reflects improved success rates but also indicates faster and more stable task completion. Overall, these results suggest that CIM consistently outperforms all baselines in both effectiveness and efficiency.

Analysis of ICM. We further analyze the role of introducing the intrinsic curiosity reward r_{ICM}. As shown in Fig. 4b, the incorporation of ICM accelerates agent learning. Moreover, Figure 8 provides additional evidence that ICM leads to an earlier increase in the number of control steps applied to the target object in each episode. This facilitates earlier and more frequent interactions between the agent and the target object, thereby promoting more efficient early-stage exploration. Consequently, the agent acquires more informative signals, enabling CCDM to achieve faster and more accurate learning.

Learned LCGs. For clarity, we visualize the local causal graphs using only the object's (x, y, z) position as the target variables, while the model takes the full state-action input (s, a). Figure 9 presents the causal graph learned in the GymFetch/PickAndPlace environment. The Hamming distance between the learned causal graph and the ground-truth graph is 26 out of 369 edges (93.0% accuracy).

Table 12: Episodic returns averaged over 100 test episodes (mean $\pm$ std) across environments.

Method	Push (↑)	PickAndPlace (↑)	RotTable (↑)	pick-place (↑)	pick-place-wall (↑)
CIM(Ours)	**-10.31** $\pm$ **0.56**	**-8.56** $\pm$ **0.53**	**-8.79** $\pm$ **0.37**	**-41.66** $\pm$ **1.61**	**-59.78** $\pm$ **3.72**
CID	-10.88 $\pm$ 0.91	-9.04 $\pm$ 0.41	-14.90 $\pm$ 2.22	-57.84 $\pm$ 6.34	-91.42 $\pm$ 8.65
DuRND	-10.38 $\pm$ 0.85	-9.09 $\pm$ 0.86	-25.56 $\pm$ 3.02	-45.48 $\pm$ 3.57	-106.81 $\pm$ 10.71
Ensemble_Dis	-14.43 $\pm$ 1.23	-8.78 $\pm$ 0.41	-52.16 $\pm$ 4.19	-71.03 $\pm$ 7.84	-250.00 $\pm$ 0.00
HER	-13.78 $\pm$ 1.18	-16.30 $\pm$ 1.88	-85.98 $\pm$ 2.42	-110.74 $\pm$ 10.25	-250.00 $\pm$ 0.00

E Implementation and Hardware Configurations

E.1 Implementation

Table 13 summarizes the network architecture of CCDM, including the encoder, decoder, and dynamics modules, as well as the key hyperparameters. Table 14 reports the main hyperparameters

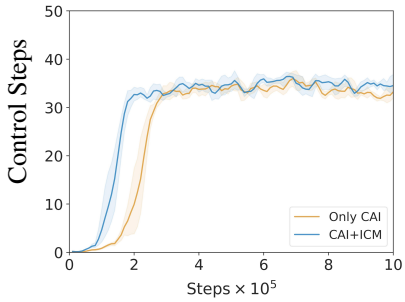


Figure 8: Target object control steps per episode.

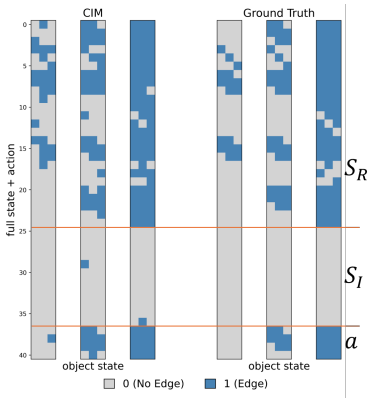


Figure 9: Comparison of the learned local causal graphs and the ground-truth graphs.

of the SAC algorithm. The same set of hyperparameters is shared across all tasks for both CCDM and SAC.

E.2 Hardware Configurations

All experiments were conducted on the machine listed in Table 15. On this setup, running our CIM algorithm for 10^6 environment steps requires roughly 12 hours (±2 hours).

Table 13: Parameters of CCDM.

Category	Parameter	Value
Training	optimizer	Adam
	learning rate	3×10^{-4}
	batch size	500
	training intervals (episodes)	200
CCDM	encoder hidden sizes	$[32, 16]$
	decoder hidden sizes	$[16, 32]$
	dynamics hidden sizes	$[256, 256]$
	activation function	ReLU
	code dimension	16
	codebook size	4
	VQ decay	0.99
	regularization coefficient λ	0.1
	commit coefficient β	0.25

Table 14: Parameters of SAC.

Category	Parameter	Value
Training	optimizer	Adam
	learning rate	3×10^{-4}
	batch size	256
	replay buffer size	10^6
SAC	policy type	Gaussian
	actor hidden sizes	$[256, 256]$
	critic hidden sizes	$[256, 256]$
	activation function	ReLU
	action squashing (actor)	Tanh
	entropy target	$-\lvert\mathcal{A}\rvert$
	discount factor γ	0.99
	soft update τ	0.005
	temperature α	learned
	target entropy	automatic
Exploration	action samples for r_{CAI} K_s	32
	r_{CAI} coefficient α_1	0.2
	r_{ICM} coefficient α_2	0.1
	HER type	future
	HER replay p	0.8

Table 15: Hardware and System.

Item	Details
Operating System	Ubuntu 18.04.2 LTS
CUDA Version	12.4
CPU Configuration	2 Sockets $\times$ 64 Cores (128 Threads)
GPU	NVIDIA GeForce RTX 3090
GPU Memory	24 GB

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately summarize our main contributions, including the proposed CCDM framework, the intrinsic reward method built upon the learned causal model, and the filtering mechanism for task-irrelevant states. The claims are consistent with the theoretical formulation and experimental results, which demonstrate the effectiveness, robustness, and out-of-distribution generalization capability of our method.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses its limitations and possible future extensions. Although CIM is effective in sparse-reward manipulation tasks, it is currently designed and evaluated mainly for state-based observations. Extending CIM to high-dimensional image inputs remains a challenge and would require additional representation learning components.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper only provides theoretical motivation and background for the method.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides the essential experimental details needed to reproduce the main results, including the task settings, model architecture, hyperparameters, and evaluation procedures.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility.

In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The code and data are not publicly available at the time of submission. We plan to release the implementation and reproduction instructions upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the key training and evaluation details required to understand the experimental results, including the task settings, training and test protocols, network architectures, hyperparameter configurations, optimizer settings, and evaluation metrics. The hyperparameters shared across tasks are also explicitly reported.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports performance averaged over 5 random seeds and includes standard deviation as error bars in the experimental results. This provides an appropriate measure of statistical variability across runs.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides information about the computational resources used for the experiments, including the type of compute hardware (GPU), training time, and key implementation details. This is sufficient to understand the computational requirements for reproducing the reported results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research fully complies with the NeurIPS Code of Ethics. The work is conducted in simulated robotic environments, does not involve human subjects, personal data, or sensitive information, and uses publicly available benchmarks. No ethical concerns are identified in terms of data collection, experimentation, or deployment.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work is intended for controlled research environments. It does not introduce direct societal risks in its current experimental setting. However, as with most reinforcement learning methods for autonomous systems, potential negative impacts may arise if deployed improperly in real-world robotic applications. The paper does not target such deployment scenarios.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: There is no such risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper uses publicly available environments and open-source baseline implementations. All external assets are properly credited and used in accordance with their licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: This paper does not release new assets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The proposed method does not involve the use of LLMs as part of the core algorithm or experiments.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.