Learning Interactive World Model for Object-Centric Reinforcement Learning

Fan Feng[1,2] **Phillip Lippe**[3*] **Sara Magliacane**[3]
[1] University of California San Diego [2] Mohamed bin Zayed University of Artificial Intelligence
[3] University of Amsterdam
{ffeng1017,phillip.lippe,sara.magliacane}@gmail.com

Abstract

Agents that understand objects and their interactions can learn policies that are more robust and transferable. However, most object-centric RL methods factor state by individual objects while leaving interactions implicit. We introduce the Factored Interactive Object-Centric World Model (FIOC-WM), a unified framework that learns structured representations of both objects and their interactions within a world model. FIOC-WM captures environment dynamics with disentangled and modular representations of object interactions, improving sample efficiency and generalization for policy learning. Concretely, FIOC-WM first learns object-centric latents and an interaction structure directly from pixels, leveraging pre-trained vision encoders. The learned world model then decomposes tasks into composable interaction primitives, and a hierarchical policy is trained on top: a high level selects the type and order of interactions, while a low level executes them. On simulated robotic and embodied-AI benchmarks, FIOC-WM improves policy-learning sample efficiency and generalization over world-model baselines, indicating that explicit, modular interaction learning is crucial for robust control[2].

1 Introduction

World models aim to learn state abstractions and action-conditioned dynamics that capture the evolution of high-dimensional observations, along with auxiliary information (e.g., rewards, skills), for decision-making tasks [1–5]. Recent advances have demonstrated their effectiveness in downstream applications, such as robotics [2, 6–10] and autonomous driving [11–14].

One of the central challenges in world model is to extract low-dimensional, structured latent representations from high-dimensional observations, which often display high complexity and variability across both semantic and dynamic aspects. On the dynamics side, latent spaces often contain underlying structures [15, 16]. Prior work imposes structural priors to learn compact latents that encourage disentanglement and capture relational or compositional patterns [17–23]. On the semantics side, pre-trained visual features are leveraged to better encode rich content and improve fidelity [9, 24–31]. Collectively, these approaches learn compressed, structured representations of high-dimensional perceptual data to support downstream decision making. However, it remains unclear to what extent such compression and structure are necessary and sufficient for down-streaming policy learning.

In this work, we study which types and degrees of decomposition structure make latent representations effective for efficient and generalizable policy learning. Real-world settings exhibit substantial variability in both visual appearance and dynamic interactions, often involving multiple objects with diverse attributes. It is therefore natural to reason in terms of *objects*, their *interactions*, and

[*] Now at Google Deepmind
[2] Project page: https://sites.google.com/view/fioc-wm.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

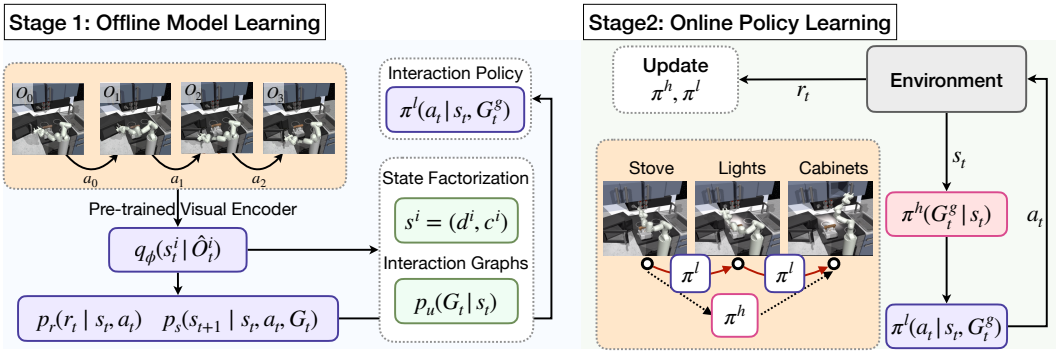

Figure 1: The overall pipeline, including offline model learning (left) and online policy learning (right) phases. The illustrative examples are from the Franka-kitchen environment [40].

the *attributes that induce these interactions*. To this end, we propose the Factored Interactive Object-Centric World Model (FIOC-WM), which learns a two-level factorization: an *object-level* representation with explicit interactions, and an *attribute-level* representation for each object. This factorization is then exploited for down-streaming planning and control.

At the *object* level, we consider both the decomposition of scenes into independently evolving objects and the modeling of their interactions. Modeling the interactions among objects is crucial for effective policy learning as real-world dynamics are heavily influenced by rich interactions among objects, such as collisions, containment, stacking, and physical forces like friction or gravity, which collectively determine the evolution of the environment [32–34].

At the *attribute* level, each object can be further factorized into attributes based on their temporal behavior, e.g., if they are static (e.g., color, shape) or dynamic (e.g., position, velocity) over time. This factorization provides a principled inductive bias to reduce redundancy and highlight the minimal sufficient components needed for planning and control. Importantly, this also supplements the accurate object-level interaction modeling as the interaction can be further factorized: for each object, only the dynamic part (e.g., position, velocities) will be changed during interactions with others. By incorporating both object-level and attribute-level factorization, we can precisely model the dynamics of all objects, including their interactions.

This structured modeling enables accurate prediction of system behavior and allows the learned interaction models to serve as efficient surrogates for decision making. Building on recent hierarchical RL with object-centric subgoals [35–37], we instantiate subgoals as object interactions, allowing complex tasks to be decomposed into sequences of interaction primitives and thereby enabling more efficient planning and control.

FIOC-WM jointly factorizes the static attributes and dynamic variables of each object in the environment, as well as their interactions with each other and the agent. After learning the FIOC-WM, we can then leverage its interaction models to learn an interaction-centric policy. This enables efficient solutions for long-horizon policy learning. Inspired by recent work [26, 27], we use pre-trained visual embeddings [38, 39] as surrogates for raw high-dimensional observations, facilitating the learning of semantically meaningful latents. FIOC-WM can recover interactions and learn the factorized states within the latent representations derived from these visual embeddings. The learned interactions are then used to train a policy designed to induce the desired interactions between objects. These offline-learned policies are subsequently employed as composable modules for long-horizon tasks. We evaluate FIOC-WM on a diverse set of robotic control and embodied AI benchmarks, demonstrating enhanced world model capability and more efficient downstream policy learning by employing the appropriate factorization and leveraging it as sub-tasks.

2 Factored Interactive Object-centric POMDP

We focus on a Partially Observable Markov Decision Process (POMDP) [41] and consider an environment in which objects interact with each other, and in which there are global latent factors that can affect or modulate these interactions. We denote the state at timestep t as $\mathbf{s}_t$ and assume it can be

factored across N objects. Moreover, we assume that the state of each object i can be represented as $\mathbf{s}_t^i = \{\mathbf{d}_t^i, \mathbf{c}^i\}$, where $\mathbf{d}_t^i$ represents the dynamic, time-varying variables (e.g., position, velocity) and $\mathbf{c}^i$ represents the constant, time-invariant properties such as color, mass, and friction, some of which can affect the dynamics of the object.

We represent interactions between objects with a sequence of time-varying graphs $\mathcal{G} = \{G_1, \ldots, G_T\}$, where each edge in a graph G_t captures an interaction between two objects at time t. This models that at each timestep, different objects might interact. We also assume that these graphs are sparse, meaning that at each timestep there are only a subset of objects interacting.

For each object, we define a self-transition function f_{self}, which represents the evolution of the object dynamics without interactions. In the self-transition function the constant properties influence the evolution of its dynamic variables over time, but not viceversa. When two objects i and j interact, an object can only affect the dynamic variables of the other object through the interaction transition function f_{inter}. More formally, we model that the state transition for object i follows the form:

$$\mathbf{d}_{t+1}^i = f_{\text{self}}(\mathbf{d}_t^i, \mathbf{c}^i, \mathbf{a}_t, \epsilon_t) + \sum_{j \in \mathcal{N}_t(i)} f_{\text{inter}}(\mathbf{d}_t^i, \mathbf{d}_t^j, \mathbf{c}^j, \delta_t),$$

where $\mathcal{N}_t(i)$ denotes the set of objects interacting with object i at time t, and ϵ_t and δ_t indicate the latent noise variables that model the stochasticity of the system.

We assume that also the observations $\mathbf{o}_t$ are factored across the N objects and that the generating process for observation of object i at time t is $\mathbf{o}_t^i = g(\mathbf{s}_t^i, \epsilon_t^i)$, where ϵ_t^i is a latent i.i.d. random noise that represents the stochasticity in the observations. Finally, as in standard settings, the reward function is a function of the global state $\mathbf{s}_t$ and the action, i.e., $r_t = h(\mathbf{s}_t, \mathbf{a}_t)$.

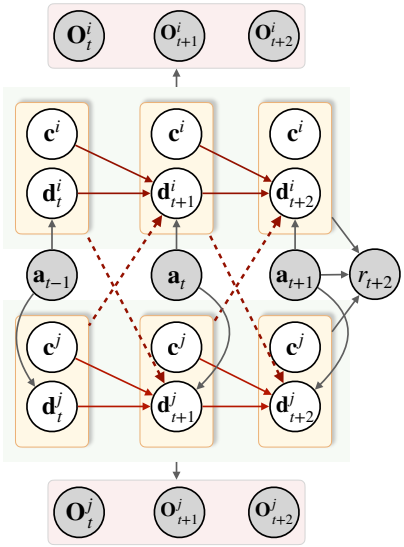

Figure 2: An example of a FIOC-POMDP, where we only show the reward for $t+2$ for clarity. Gray nodes are observed variables, while white nodes are latent variables. Each orange box represents the state of an object (in this case, objects i and j). Red solid edges are the state transition per object, and dashed edges are the interactions among objects.

We call a model that satisfies all of these assumptions a Factored Interactive Object-centric POMDP (FIOC-POMDP). Fig. 2 depicts an example of a FIOC-POMDP.

3 Learning the FIOC World Models

The overall framework (Fig. 1) consists of two stages: (1) offline model learning (Fig. 3) and (2) hierarchical policy learning. In offline model learning, we learn a world model for a FIOC-POMDP as two-level factorization of the latent space, at the object and attribute levels, and model latent dynamics based on object interactions. Leveraging the learned interactions, we train an inverse dynamics model to map the states of two separate objects to the states where they interact effectively, which we use as an interaction policy. In hierarchical policy learning, a hierarchical policy is trained. The high-level policy selects a sequence of target interaction graphs, while the low-level interaction policy trained in the first stage executes them by inducing the corresponding interaction graph in the environment.

3.1 Stage 1: Offline Model Learning

We encode the observations using object-centric representation learning built on top of pre-trained models such as DINO-v2 [38] and R3M [39], which have been empirically shown to provide high-quality image understanding capabilities [42–44, 27, 45] and facilitate robotic manipulation tasks [25, 26, 31]. Building on the empirical and theoretical work regarding the recoverability of latent features from supervised pre-trained models [46], we assume that these embeddings provide sufficient features and information for world models. This includes supporting the dynamics and reward models, as well as capturing action-related features effectively. Then, similarly to Zadaianchuk

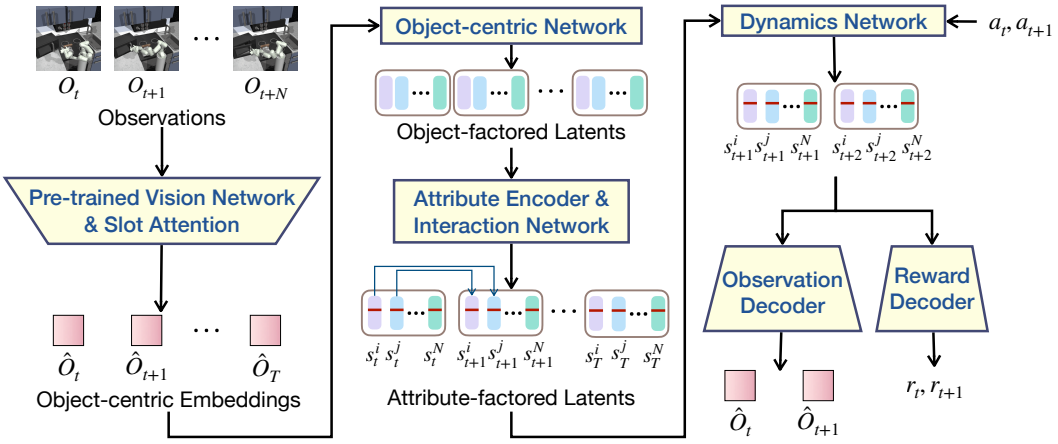

Figure 3: The pipeline of Offline Model Learning (Stage 1) jointly learns the observation function, state factorization, dynamics model, and reward model. Although Fig. 2 includes low-level policy learning as part of Stage 1, for clarity, we defer the discussion of low-level policy learning to Stage 2.

et al. [47], we use slot attention [48] to cluster the object-centric representation on top of the embeddings. The slot attention outputs a set of slot representations, which we use as the factored observation $\{\hat{\mathbf{o}}^1, \hat{\mathbf{o}}^2, \ldots, \hat{\mathbf{o}}^N\}$ corresponding to the factored raw observation $\{\mathbf{o}^1, \mathbf{o}^2, \ldots, \mathbf{o}^N\}$. To map these factored observations to factored states $\{\mathbf{s}^1, \mathbf{s}^2, \ldots, \mathbf{s}^N\}$, we train a variational auto-encoder (VAE) [49] with the encoder $q_\phi(\mathbf{s}^i|\hat{\mathbf{o}}^i)$ and the decoder $p_\psi(\hat{\mathbf{o}}^i|\mathbf{s}^i)$, where $\mathbf{s}^i$ is the latent state corresponding to the observation $\hat{\mathbf{o}}^i$, and the shared parameters are used across all slots.

To encourage structured representations, we learn to factorize the latent state into static and dynamic components, denoted by $\mathbf{c}$ and $\mathbf{d}$, respectively. Two separate encoders, $f_c(\mathbf{s})$ and $f_d(\mathbf{s})$, are used to extract static and dynamic features from observations. We assume that static features remain invariant over time, while dynamic features evolve. To enforce this, we regularize the output of $f_c(\mathbf{s})$ to remain temporally consistent for each of the N object slots:

$$\mathcal{L}_{\text{static}} = \sum_{t=1}^{T-1} \sum_{i=1}^{N} \left| f_c(\mathbf{s}_{t+1}^i) - f_c(\mathbf{s}_t^i) \right|^2, \tag{1}$$

where T is the number of time steps. To ensure that different objects encode distinct static attributes, we use a contrastive loss [50] that separates static features across slots:

$$\mathcal{L}_{\text{con}} = -\sum_{t=1}^{T-1} \sum_{i=1}^{N} \log \frac{g\left(f_c(\mathbf{s}_t^i), f_c(\mathbf{s}_{t'}^i)\right)}{g\left(f_c(\mathbf{s}_t^i), f_c(\mathbf{s}_{t'}^i)\right) + \sum_{j \in \mathcal{N}} g\left(f_c(\mathbf{s}_t^i), f_c(\mathbf{s}_{t'}^j)\right)}, \tag{2}$$

where t' is a different time step and $\mathcal{N}$ denotes a set of negative slots $j \neq i$ from the same scene. g is the distance measurement of the representation, we use cosine similarity here.

For the dynamic features, we leverage their temporal evolution to model latent state transitions, as only $\mathbf{d}$ varies over time. We adopt the variational inference framework [49] to learn the encoder $f_d(\mathbf{s})$, parameterized by a GRU [51], which captures the dynamics of each object slot. The prior over the dynamic state $\mathbf{d}_t$ is factorized across the N object slots as: $p_s(\mathbf{d}_t \mid \mathbf{d}_{t-1}, \mathbf{a}_{t-1}) = \prod_{i=1}^{N} p_s(\mathbf{d}_t^i \mid \mathbf{d}_{t-1}, \mathbf{a}_{t-1}, G_t)$, where G_t denotes the interaction graph representing the relational structure among objects at time t. In other words, G_t captures the pairwise interactions between objects at time step t, where each edge indicates whether an interaction exists between a pair of objects. Concretely, this is represented as a binary adjacency matrix of size $N \times N$, where N is the number of objects.

The posterior over $\mathbf{s}_t$ is conditioned on the current the visual embeddings $\hat{\mathbf{o}}_t$ and the hidden state $\mathbf{h}_t = \text{GRU}(\mathbf{s}_{t-1}, \mathbf{h}_{t-1})$, as: $q_\phi(\mathbf{s}_t \mid \hat{\mathbf{o}}_t, \mathbf{h}_t)$. Then we use an observation decoder to reconstruct observations: $p_\sigma(\hat{\mathbf{o}}_t \mid \mathbf{s}_t)$, with the reconstruction loss:

$$\mathcal{L}_{\text{recon}} = \sum_{t=1}^{T} \left\| \hat{\mathbf{o}}_t - \hat{\mathbf{o}}_t^{\text{decoded}} \right\|^2, \tag{3}$$

where $\hat{\mathbf{o}}_t^{\text{decoded}}$ is sampled from $p_\sigma(\cdot \mid \mathbf{s}_t)$. To capture temporal consistency, we also predict the next-step observation:

$$\mathcal{L}_{\text{pred}} = \sum_{t=1}^{T} \left\| \hat{\mathbf{o}}_{t+1} - \hat{\mathbf{o}}_{t+1}^{\text{decoded}} \right\|^2 . \tag{4}$$

We encourage alignment between the posterior and the prior using KL divergence:

$$\mathcal{L}_{\text{KL}} = \sum_{t=1}^{T} \text{KL}\left(q_\phi(\mathbf{s}_t \mid \hat{\mathbf{o}}_t, \mathbf{h}_t) \,\|\, p_s(\mathbf{s}_t \mid \mathbf{s}_{t-1}^s, \mathbf{a}_{t-1}, G_t) \right) . \tag{5}$$

Similarly, we apply a reward decoder $p_r(r_t \mid \mathbf{s}_t, \mathbf{a}_t)$ based on the learned latent states and actions. The reward loss is as follows:

$$\mathcal{L}_{\text{rew}} = \sum_{t=1}^{T} \left\| \hat{r}_t - r_t \right\|^2 . \tag{6}$$

To learn the interaction graph G_t, we use the current estimated latent states $\mathbf{s}_t$ as input. We introduce a surrogate latent variable $\mathbf{u}_t$ that parameterizes the distribution over interaction graphs. This captures the underlying interactions that may vary over time.

Specifically, for each object pair (i, j) at time t, we encode their latent states s_t^i and s_t^j using a GRU encoder to obtain a pairwise embedding:

$$\mathbf{u}_t^{ij} = f_{\text{enc},\phi_u}(\mathbf{s}_t^i, \mathbf{s}_t^j) \tag{7}$$

The transition of $\mathbf{u}_t$ is modeled as: $p_u(\mathbf{u}_t \mid \mathbf{s}_t) = f_u(\mathbf{s}_t)$, where f_u is a parameterized function that captures the dependencies among the current latent states $\mathbf{s}_t$. We consider two approaches for learning the state transition distribution p_s: (i) learning variational masks, following [52, 53]; and (ii) applying conditional independence testing, following [54]. The detailed loss functions are provided in Appendix C.2.

3.2 Stage 2: Online Hierarchical Policy Learning

In this section, we describe how we use the learned interactive world model for object-centric RL, particularly for long-horizon task learning. Our framework is built on the recent work that models the object interactions as skills [37]. The key intuition is that long-horizon tasks can be decomposed into a sequence of interactions.

Our approach first focuses on learning a *low-level policy* capable of invoking the desired interactions. Based on the learned interactive world model, we can accurately predict the dynamics of interactions and the regimes governing these interactions. This enables the agent to learn the policy by leveraging the predicted interactions to learn the inverse mapping from interactions to actions. We learn the low-level policy π^l by employing model predictive control (MPC) [55, 56, 5] or proximal policy optimization (PPO) [57], where the initial and target interaction of two objects are provided. At time step t, we are given the target interaction graph at future steps from high-level policy, denoted as G_t^g, and the low-level policy is $\pi^l(\mathbf{a}_t \mid \mathbf{s}_t, G_t^g)$. Given the learned transition models p_s and p_u, we use $\mathbf{s}^i$ and $\mathbf{u}^g$ to infer the target states $\mathbf{s}_g^i$ and $\mathbf{s}_g^j$. Using these inferred target states, we apply MPC or PPO to generate a sequence of actions that transitions the system from t to $t + k$ while minimizing the discrepancy between the predicted and target states. We learn the low-level policy during world model learning (Stage 1), and then fine-tune it with online data during Stage 2, where the policy is updated each time new interaction data becomes available.

We then learn a schedule of interactions for the model to handle long-horizon tasks by optimizing the task reward. We learn the chain-of-interactions for the high-level policy $\pi^h : \mathcal{S} \to \mathcal{G}$, which selects the interaction graph $\mathcal{G}$ based on the input state $\mathcal{S}$. This implies that the action space corresponds to graph selection, but this space can grow exponentially with the number of objects. To address this, following previous works on skill discovery with object interaction [58, 37], we impose constraints by limiting the number of objects considered at each time step. Following the graph selection policy introduced in [37], at any given time, we focus on a fixed subset of objects (smaller than 2), leveraging a diversity reward r_{div} as a surrogate to make the selection process diverse. We define $r_{\text{div}} = 1/\sqrt{|G_{\text{visited}}|}$, where $|G_{\text{visited}}|$ is the number of graphs that have been visited in the past transitions. Then the high-level policy $\pi^h(G_t^g \mid \mathbf{s}_t)$ is updated with both the task reward r_{task} and this diversity reward r_{div}.

3.2.1 Practical Implementation

We assume that each state $\mathbf{s}_t$ is associated with an interaction graph G^t, and the final task corresponds to reaching a desired target graph G^g. The high-level policy π^h selects a sequence of intermediate *subgoal graphs* that gradually transform G^t into G^g, where each subgoal graph differs from the previous one in only a single interaction. For example, in a task such as moving a kettle from the counter to the stove, the graph transitions involve first enabling an interaction between the arm and the kettle, followed by an interaction between the kettle and the stovetop.

To make the subgoal selection both tractable and structured, we do not sample directly from the full space of possible object interactions. Instead, at each decision point, we first identify a small subset of objects (typically one or two objects) as primary candidates for initiating interaction changes. These candidates define the anchor object(s) i, and we then select a target object j conditioned on i to form the proposed subgoal interaction (i, j). This scheme reduces the combinatorial action space and leads to more localized graph transitions. Note that the selected subset does not constrain the interaction to only occur between these objects; rather, it defines a focused region of the graph for subgoal exploration.

4 Related Work

Our framework aims to uncover interactive and factored object-based representations of environments, so it is closely related to factored RL, particularly object-centric RL. Factored RL models the environment in terms of Factored Markov Decision Process [59], where the state of the Markov decision process is factored in state components and sparse relationships exist among state components, actions, and rewards. This factorization enables efficient policy solutions [60, 61]. A specific type of factorization, which we also adopt, is object-centric reinforcement learning [62], where states or observations are grouped into object-centric clusters. In object-centric RL, actions typically target only a subset of objects, and rewards are often associated with the states of specific objects or object subsets. This facilitates more structured and efficient decision-making.

Recent works on object-centric RL can be broadly categorized into two major directions: (i) learning object-centric representation and (ii) modeling the object relations and policy architectures for compositional generalization. For the first line of research, approaches focus on using object-centric representation learning techniques [48, 63–65] to extract meaningful object-level features from raw observations. These methods then learn object-centric policies directly from object-centric representations [66, 67, 62]. The second line of work develops object-centric policies by modeling object relations and policy structures, incorporating inductive biases in the state transition and policy networks. Methods include the use of graph neural networks [68], linear relational networks [69], self-attention, and deep sets [70]. The learned object-centric states and relational structures are then used to achieve compositional generalization in reinforcement learning [71, 18, 33, 72, 22, 73]. Our work combines ideas from both directions, especially related to the series of works [33, 71, 22, 67], which learn factored state attributes, providing a more fine-grained representation than object-centric factorizations, and also model the interactions among objects to achieve compositional generalization. However, we go beyond object-centric policies by learning an interaction-centric policy.

Our framework, which uses low-level and high-level policy for decomposing complex tasks into interaction learning, is similar to hierarchical reinforcement learning (HRL). HRL typically consists of a high-level policy (often referred to as an option [74], sub-skills, or sub-goals in the literature) and a low-level policy, enabling the efficient learning of complex RL tasks [75]. Within the scope of HRL, the most close to our work is the line of research that focuses on learning goal-conditioned hierarchical policies or hierarchical skill discovery. Zadaianchuk et al. [66] propose the goal-conditioned hierarchical policies with learning object-based hierarchical goals. Hierarchical skill discovery focuses on decomposing complex tasks into object-wise or object-interaction-based components. For instance, Wang et al. [37] use conditional independence testing to identify sub-goals, while Chuck et al. [35, 73] and Hu et al. [76] use Granger causality or counterfactual reasoning to uncover hierarchical structures. Our work is also built upon those works in using interactions as sub-skills [36, 37], but we learn interaction models jointly with observations and dynamics within the world model directly from high-dimensional inputs.

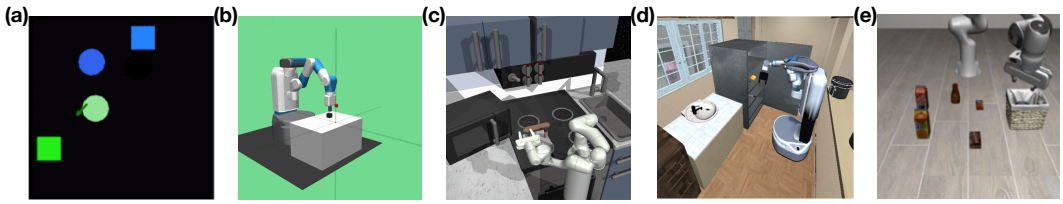

Figure 4: Visualization of evaluated benchmarks: (a). Sprites-World; (b). OpenAI-Gym Fetch; (c). Franka Kitchen; (d). i-Gibson; and (e). Libero. A larger version is in Table A4.

5 Experiments

To evaluate the effectiveness of our proposed interactive world model and policy learning framework, we aim to address the following questions: (i) *How accurately does the model learns the state disentanglement and interaction models?* (ii) *How well does it perform in long-horizon task learning?* and (iii) *How well does the framework achieve compositional generalization?*

To answer these questions, we evaluate our method on a range of simulated control, robotic manipulation, and embodied AI benchmarks, including SpritesWorld [77], OpenAI-Gym Fetch [78], iGibson [79], and Libero [80]. We consider both reinforcement learning and imitation learning tasks.

Baselines. For online RL or imitation learning, we compare against established baselines including DreamerV3 [4], TD-MPC2 [81] and the object-centric model-free method EIT [67]. For offline RL, we compare with DINO-WM [27], which also leverages DINO-based pretraining for downstream planning.

Benchmarks. We consider long-horizon tasks that require completing several sub-skills to achieve the overall objective. OpenAI Gym Fetch [78] is a simulated environment featuring a Fetch robotic arm capable of manipulating cubes and switches. The tasks involve completing sub-tasks that require pushing or switching a varying number of objects. Franka-kitchen [40] is an environment where the 7-DoF Franka Emika Panda arm performs tasks in a kitchen. We consider several sequential sub-tasks, such as turning on the microwave, moving the kettle, turning on the stove, and turning on the light. i-Gibson [82] is a simulated environment with a Fetch robot operating in everyday household tasks with rich objects and interactions. Similarly to [37], we consider the tasks with the peach object. Libero [80] is a benchmark for lifelong robot learning and imitation learning in household and tabletop environments. We focus on randomly selected tasks within libero-goal.

5.1 Evaluation Metrics.

In addition to evaluating policy learning and planning performance, we assess the effectiveness of world model learning by examining three key aspects: observation and dynamics modeling, interaction learning, and disentanglement quality. Specifically, for all methods (excluding those evaluated under nSHD), we adopt variational masks to infer the interaction structures. For downstream control, we apply MPC for Gym-Fetch and Franka-Kitchen, and use PPO for LIBERO and iGibson.

Observation and Dynamics Modeling We measure the predictive quality of future observations using the Learned Perceptual Image Patch Similarity (LPIPS) metric [83], which evaluates perceptual similarity between predicted and ground-truth image patches.

Interaction Learning We evaluate the ability of our model to learn interactions through two approaches: (i) *Variational mask learning*, where the state encodes adjacency matrices as latent variables. Each edge is sampled from either a differentiable approximation of a categorical distribution [84, 85, 52] or a discrete codebook [53]. (ii) *Conditional independence testing*, where we test for the existence of interaction using parametric models to predict dynamics [54]. We compare our approach with baselines that do not explicitly model dynamic structure, but instead rely on post hoc analysis based on attention weights to infer interactions, as in local causal discovery methods [86, 87]. We use normalized Structured Hamming Distance (nSHD). Further details are provided in Appendix C.2.

Environment	Dreamer-V3	TD-MPC2	EIT	DINO-WM	FIOC
Fetch	0.042	0.039	0.026	0.009	**0.007**
Kitchen	0.102	0.123	0.096	**0.035**	0.038
Libero	0.089	0.061	0.040	0.035	**0.027**

Table 1: Comparison of world models on LPIPS metrics on Fetch, Kitchen, and Libero.

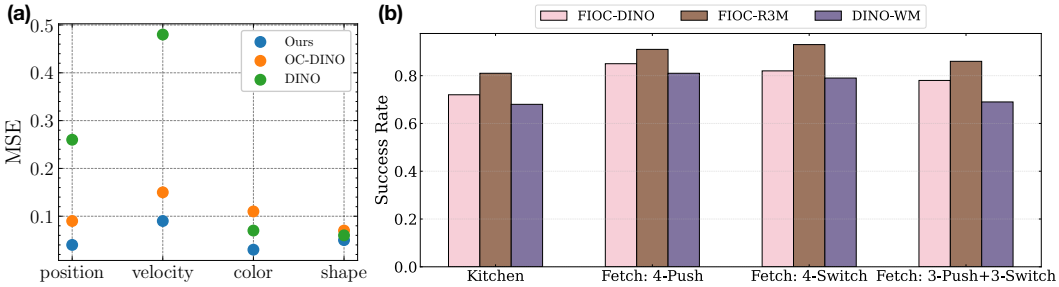

Figure 5: (a) Evaluation of state factorization in Sprites-world. We report MSE from linear probing to assess the quality of the learned representations against ground-truth attributes.(b) Offline RL performance (success rate) comparison with DINO-WM, including FIOC-DINO and FIOC-R3M.

Disentanglement Quality In SpritesWorld [77], we perform a linear probing analysis by training a linear regression layer on top of the learned representations to predict ground-truth static and dynamic factors. Static factors include object color and shape (encoded as one-hot vectors), while dynamic factors consist of object positions and velocities.

Policy Learning We consider both the policy performance on single-task and the generalization task. For generalization tasks, we consider three types of generalization: (1) *Attribute Generalization*: we evaluate for zero-shot generalization on new composition of object attributes; (2) *Object Attribute Composition*: we train models on domains with specific combinations of object attributes (e.g., color, shape, or material) and test them on domains with unseen attribute combinations; and *Skill Composition Generalization*: we train models on tasks with simple combinations of skills and test them on tasks requiring new combinations of skills. For all tasks, we use the average success rate as the evaluation metric.

5.2 Results on Learning World Models.

As evaluation of the learned dynamics and observations, Table 1 reports the LPIPS metric (Full Results are in Table A3). Compared to the baselines, our method achieves comparable or better reconstruction performance, particularly on the Fetch and Libero environments, where object interactions and dynamics are complex. Full results are in Appendix D.1.

We report also results on the accuracy of the learned interactions, quantified by the normalized Structured Hamming Distance (SHD) between the inferred interaction structures $\hat{\mathcal{G}} = \{\hat{G}_1, \ldots, \hat{G}_T\}$ and the ground truth structures. Fig. 6(a) presents the results on attribute and compositional generalization. For each bar, the shaded areas represent the performance of single-task learning with the same number of objects. The gap between the top of each bar and the top of the corresponding shaded area quantifies the performance drop when generalizing to novel scenarios (i.e., *empirical generalization gap*). These results demonstrate that FIOC consistently outperforms attention-based methods across all cases, verifying the importance of explicitly modeling the interaction structures and their changes using regime variables. And importantly, FIOC demonstrates superior generalization compared to attention-based methods, as shown by the smaller empirical generalization gap. Among the three versions of FIOC, all achieve strong attribute-level and compositional generalization. Notably, the variational masks with categorical distributions perform best, particularly in scenarios with a large number of objects. Full results are in Appendix D.1.

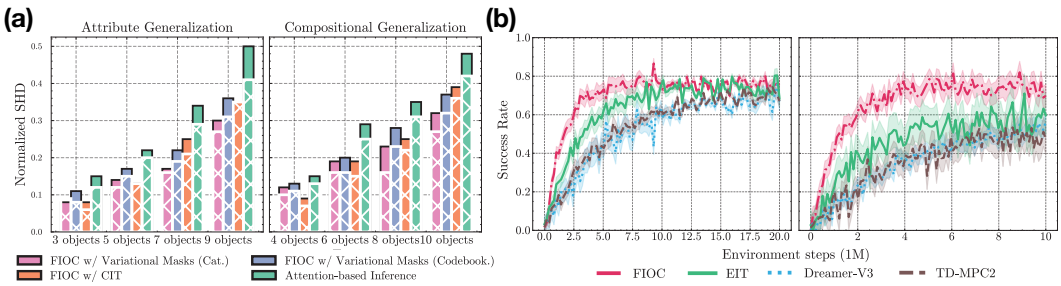

Figure 6: (a) Evaluation of learned interaction graphs with Normalized SHD for attribute and compositional generalization on Sprites-World with multiple objects. The shaded areas show results with the same number of objects for single-task learning. Lower values mean better performance. (b) Learning curves of single-task learning for i-Gibson (left) and Libero (right).

Fig. 5(a) reports the linear probing MSE of the learned static and dynamic representations, $\mathbf{c}$ and $\mathbf{d}$, against ground-truth attributes in the Sprites-world environment. We evaluate both the DINO-v2 raw input features and the object-centric DINO features (obtained from our first-stage learning without disentanglement). Our method achieves the best factorization of attributes: $\mathbf{c}$ and $\mathbf{d}$ effectively capture useful representations for dynamic features (i.e., position & velocity) and static features (i.e., color & shape), respectively. Notably, the object-centric DINO generally outperforms vanilla DINO on dynamic features. However, object-centric clustering tends to degrade static attribute representations such as color and shape. Our disentanglement module addresses this limitation by improving the representation of static attributes within each object.

	Envs	FIOC	Dreamer-V3	EIT	TD-MPC2
Attri. Gen.	Push & Switch	0.91± 0.05	0.90± 0.07	0.92± 0.04	**0.95**± 0.02
	i-Gibson	**0.79**± 0.13	0.62± 0.16	0.70± 0.14	0.65± 0.15
	Libero	**0.76**± 0.14	0.59± 0.18	0.73± 0.12	0.69± 0.18
Comp. Gen.	Push & Switch	**0.86**± 0.10	0.81± 0.12	0.83± 0.02	0.79± 0.08
	Libero	**0.70**± 0.09	0.58± 0.12	0.65± 0.08	0.63± 0.14
Skill Gen.	Push & Switch	**0.81**± 0.06	0.66± 0.10	0.73± 0.08	0.65± 0.13
	Franka Kitchen	**0.73**± 0.06	0.59± 0.09	0.65± 0.18	0.62± 0.08

Table 2: Policy learning (success rate) of world model in Gym Fetch, Franka Kitchen, i-Gibson, and Libero tasks.

5.3 Results on Policy Learning.

Fig. 6(b) presents the learning curves (sampled every 100 time steps) on the i-Gibson and Libero tasks. The results indicate that world models incorporating object interactions, such as FIOC and EIT, achieve faster convergence compared to state-of-the-art methods like Dreamer-V3 and TD-MPC2. FIOC not only converges faster than EIT but also achieves a higher final success rate on Libero.

Fig.5(b) presents the offline RL results, comparing our method with DINO-WM [27], along with two variants of FIOC that use DINO-v2 [38] and R3M [39] as pre-trained visual embeddings. The results demonstrate that our approach achieves superior performance in both single-task learning and generalization, highlighting the advantages of the proposed two-level factorization on top of pre-trained visual features and the use of a hierarchical policy.

Table 2 presents the results of policy learning on single tasks, as well as those in the context of attribute, compositional, and skill generalization. The results indicate that FIOC performs comparably or better than other baselines in single-task learning scenarios and consistently outperforms them in all generalization tasks. Among the baselines, EIT achieves the second-best performance across generalization tasks. Detailed task settings are provided in Appendix F. The full results are in Table A2 and Fig. A3 in the appendix.

5.4 Ablation Studies

To evaluate the effectiveness of different components in both offline world model learning and online policy learning, we conduct a series of ablation studies on the following aspects. For the world model part, we consider cases: *Without state factorization:* The state s is not factorized into static and dynamic components. Instead, the state transition P_s is learned directly on the original s obtained from the DINO embeddings. *Without interaction modeling:* Instead of modeling dynamic interactions, we assume a fully connected graph for all time steps and learn the dynamics using this dense graph; and *Using random actions in offline learning:* Instead of using pre-trained policies, we train the model with random actions in the offline learning phase. For the policy learning stage, we consider the cases: *Without hierarchical policy:* Policy learning is performed directly on low-level actions without a high-level policy governing the sequence of interactions. *Without pre-trained low-level policy:* The low-level policy π^l is not trained during the offline phase but learned from scratch in the online phase. *Without diversity term:* The diversity term in high-level policy learning is disabled.

Ablations	Success Rate	
	Single Task	Comp. Gen.
FIOC	0.81	0.70
w/o Factorization	0.77($\downarrow$ 0.04)	0.64($\downarrow$ 0.06)
w/o Interaction	**0.63($\downarrow$ 0.18)**	0.52($\downarrow$ 0.18)
w/ random actions	0.64($\downarrow$ 0.17)	**0.48($\downarrow$ 0.22)**
w/o hierarchical policy	**0.58($\downarrow$ 0.23)**	**0.42($\downarrow$ 0.28)**
w/o pre-trained π^l	0.69($\downarrow$ 0.12)	0.59($\downarrow$ 0.11)
w/o diversity	0.62($\downarrow$ 0.19)	0.50($\downarrow$ 0.20)

Table 3: Ablation studies on Libero, evaluating the impact of removing specific components from the world model and policy learning. The **bold** entries indicate the ones with the largest performance drop. Light yellow and green areas represent the world model and policy learning components.

The results in Table 3 show that for world models, interaction modeling and using the pre-trained policies in offline learning are the most critical components, as their removals lead to the most significant drop in policy learning performance. For policy learning, the hierarchical policy plays the most essential role. Other components, such as state factorization, utilizing pre-trained policies instead of random actions in the offline learning phase, pre-training the low-level policy, and incorporating diversity, also contribute to improving the policy learning performance.

6 Conclusions and Discussion

We study which types and degrees of decomposition make latent representations effective for efficient and generalizable policy learning. To this end, we introduce the Factored Interactive Object-Centric World Model (FIOC-WM), which learns a two-level factorization: an object-level representation with explicit interactions and an attribute-level representation for each object. FIOC-WM learns these decomposed structures directly from observations and leverages the resulting composable interaction primitives to enhance planning and policy learning via a hierarchical RL approach. The framework exhibits strong compositional generalization across attributes, objects, and skills, demonstrating that explicit, object-centric interaction decomposition is a key inductive bias for robust control.

Limitations and Future Works FIOC-WM still relies on a pretrained object-centric model for object discovery, and its interaction models primarily generalize to seen object categories. Addressing these limitations and extending the framework to real-world robotic settings is part of the future work, potentially leveraging recent advances in robot-learning foundation models [88–97].

Acknowledgment

We would like to thank the anonymous reviewers for their helpful comments and suggestions during the review process. We also acknowledge the computational support provided by the IVI servers at the University of Amsterdam and the University HPC centers at City University of Hong Kong.

References

[1] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems (NeruIPS)*, 31, 2018.

[2] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

[3] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=0oabwyZbOu.

[4] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[5] Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*, 2022.

[6] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning (ICML)*, pages 2555–2565. PMLR, 2019.

[7] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on Robot Learning (CoRL)*, pages 2226–2240. PMLR, 2023.

[8] Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination. *arXiv preprint arXiv:2404.12377*, 2024.

[9] Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=sFyTZEqmUY.

[10] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[11] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

[12] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *European Conference on Computer Vision*, pages 55–72. Springer, 2024.

[13] Lloyd Russell, Anthony Hu, Lorenzo Bertoni, George Fedoseev, Jamie Shotton, Elahe Arani, and Gianluca Corrado. Gaia-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*, 2025.

[14] Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. *arXiv preprint arXiv:2412.03572*, 2024.

[15] Aditya Mohan, Amy Zhang, and Marius Lindauer. Structure in deep reinforcement learning: A survey and open problems. *Journal of Artificial Intelligence Research*, 79:1167–1236, 2024.

[16] Klemen Kotar, Wanhee Lee, Rahul Venkatesh, Honglin Chen, Daniel Bear, Jared Watrous, Simon Kim, Khai Loong Aw, Lilian Naing Chen, Stefan Stojanov, et al. World modeling with probabilistic structure integration. *arXiv preprint arXiv:2509.09737*, 2025.

[17] Thomas Kipf, Elise Van der Pol, and Max Welling. Contrastive learning of structured world models. *arXiv preprint arXiv:1911.12247*, 2019.

[18] Linfeng Zhao, Lingzhi Kong, Robin Walters, and Lawson LS Wong. Toward compositional generalization in object-oriented world modeling. In *International Conference on Machine Learning (ICML)*, pages 26841–26864. PMLR, 2022.

[19] Tongzhou Wang, Simon Du, Antonio Torralba, Phillip Isola, Amy Zhang, and Yuandong Tian. Denoised mdps: Learning world models better than the world itself. In *International Conference on Machine Learning*, pages 22591–22612. PMLR, 2022.

[20] Atharva Sehgal, Arya Grayeli, Jennifer J. Sun, and Swarat Chaudhuri. Neurosymbolic grounding for compositional world models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4KZpDGD4Nh.

[21] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. 2023.

[22] Fan Feng and Sara Magliacane. Learning dynamic attribute-factored world models for efficient multi-object reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2023.

[23] Junyeob Baek, Yi-Fu Wu, Gautam Singh, and Sungjin Ahn. Dreamweaver: Learning compositional world models from pixels. *arXiv preprint arXiv:2501.14174*, 2025.

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

[25] Kaylee Burns, Zach Witzel, Jubayer Ibn Hamid, Tianhe Yu, Chelsea Finn, and Karol Hausman. What makes pre-trained visual representations successful for robust manipulation? *arXiv preprint arXiv:2312.12444*, 2023.

[26] Zichen Cui, Hengkai Pan, Aadhithya Iyer, Siddhant Haldar, and Lerrel Pinto. Dynamo: In-domain dynamics pretraining for visuo-motor control. *Advances in Neural Information Processing Systems*, 37:33933–33961, 2024.

[27] Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

[28] Yucen Wang, Rui Yu, Shenghua Wan, Le Gan, and De-Chuan Zhan. Founder: Grounding foundation models in world models for open-ended embodied decision making. In *Forty-second International Conference on Machine Learning*, 2025.

[29] Calvin Luo, Zilai Zeng, Yilun Du, and Chen Sun. Solving new tasks by adapting internet video knowledge. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=p01BR4njlY.

[30] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

[31] Federico Baldassarre, Marc Szafraniec, Basile Terver, Vasil Khalidov, Francisco Massa, Yann LeCun, Patrick Labatut, Maximilian Seitzer, and Piotr Bojanowski. Back to the features: Dino as a foundation for video world models. *arXiv preprint arXiv:2507.19468*, 2025.

[32] Daniel Bear, Elias Wang, Damian Mrowca, Felix Jedidja Binder, Hsiao-Yu Tung, RT Pramod, Cameron Holdaway, Sirui Tao, Kevin A Smith, Fan-Yun Sun, et al. Physion: Evaluating physical prediction from vision in humans and machines. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[33] Akihiro Nakano, Masahiro Suzuki, and Yutaka Matsuo. Interaction-based disentanglement of entities for object-centric world models. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=JQc2VowqCzz.

[34] Shiqian Li, Kewen Wu, Chi Zhang, and Yixin Zhu. I-phyre: Interactive physical reasoning. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[35] Caleb Chuck, Kevin Black, Aditya Arjun, Yuke Zhu, and Scott Niekum. Granger-causal hierarchical skill discovery. *arXiv e-prints*, pages arXiv–2306, 2023.

[36] Caleb Chuck, Kevin Black, Aditya Arjun, Yuke Zhu, and Scott Niekum. Granger causal interaction skill chains. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=iA2KQyoun1.

[37] Zizhao Wang, Jiaheng Hu, Caleb Chuck, Stephen Chen, Roberto Martín-Martín, Amy Zhang, Scott Niekum, and Peter Stone. Skild: Unsupervised skill discovery guided by factor interactions. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[38] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

[39] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[40] Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In *Proceedings of the Conference on Robot Learning (CoRL)*. PMLR, 2020.

[41] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[42] Norman Di Palo and Edward Johns. Dinobot: Robot manipulation via retrieval and alignment with vision foundation models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[43] Yingdong Hu, Renhao Wang, Li Erran Li, and Yang Gao. For pre-trained vision models in motor control, not all policy learning methods are created equal. In *International Conference on Machine Learning (ICML)*, pages 13628–13651. PMLR, 2023.

[44] Wei-Di Chang, Francois Hogan, David Meger, and Gregory Dudek. Generalizable imitation learning through pre-trained representations. *arXiv preprint arXiv:2311.09350*, 2023.

[45] Xingyu Lin, John So, Sashwat Mahalingam, Fangchen Liu, and Pieter Abbeel. Spawnnet: Learning generalizable visuomotor skills from pre-trained network. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4781–4787. IEEE, 2024.

[46] Patrik Reizinger, Alice Bizeul, Attila Juhos, Julia E Vogt, Randall Balestriero, Wieland Brendel, and David Klindt. Cross-entropy is all you need to invert the data generating process. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=hrqNOxpItr.

[47] Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Object-centric learning for real-world videos by predicting temporal feature similarities. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=t1jLRFvBqm.

[48] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems (NeurIPS)*, 33: 11525–11538, 2020.

[49] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *The International Conference on Learning Representations (ICLR)*, 2014.

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

[51] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[52] Sindy Löwe, David Madras, Richard Zemel, and Max Welling. Amortized causal discovery: Learning to infer causal graphs from time-series data. In *Conference on Causal Learning and Reasoning*, pages 509–525. PMLR, 2022.

[53] 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. In *Proceedings of the 41th International Conference on Machine Learning (ICML)*, 2024.

[54] Zizhao Wang, Xuesu Xiao, Zifan Xu, Yuke Zhu, and Peter Stone. Causal dynamics learning for task-independent state abstraction. *International Conference on Machine Learning (ICML)*, 2022.

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

[56] KS Holkar and Laxman M Waghmare. An overview of model predictive control. *International Journal of control and automation*, 3(4):47–63, 2010.

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

[58] Zizhao Wang, Jiaheng Hu, Peter Stone, and Roberto Martín-Martín. ELDEN: Exploration via local dependencies. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=sL4pJBXkxu.

[59] Carlos Guestrin, Daphne Koller, Ronald Parr, and Shobha Venkataraman. Efficient solution algorithms for factored mdps. *Journal of Artificial Intelligence Research*, 19:399–468, 2003.

[60] Karina Valdivia Delgado, Scott Sanner, and Leliane Nunes De Barros. Efficient solutions to factored mdps with imprecise transition probabilities. *Artificial Intelligence*, 175(9-10): 1498–1527, 2011.

[61] Ian Osband and Benjamin Van Roy. Near-optimal reinforcement learning in factored mdps. *Advances in Neural Information Processing Systems (NIPS)*, 27, 2014.

[62] Jaesik Yoon, Yi-Fu Wu, Heechul Bae, and Sungjin Ahn. An investigation into pre-training object-centric representations for reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 40147–40174. PMLR, 2023.

[63] Ziyi Wu, Jingyu Hu, Wuyue Lu, Igor Gilitschenski, and Animesh Garg. Slotdiffusion: Object-centric generative modeling with diffusion models. *Advances in Neural Information Processing Systems*, 36:50932–50958, 2023.

[64] Jindong Jiang, Fei Deng, Gautam Singh, and Sungjin Ahn. Object-centric slot diffusion. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2023.

[65] Jindong Jiang, Fei Deng, Gautam Singh, Minseung Lee, and Sungjin Ahn. Slot state space models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024. URL https://openreview.net/forum?id=BJv1t4XNJW.

[66] Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Self-supervised visual reinforcement learning with object-centric representations. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=xppLmXCbOw1.

[67] Dan Haramati, Tal Daniel, and Aviv Tamar. Entity-centric reinforcement learning for object manipulation from pixels. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=uDxeSZ1wdI.

[68] Richard Li, Allan Jabri, Trevor Darrell, and Pulkit Agrawal. Towards practical multi-object manipulation using relational reinforcement learning. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 4051–4058. IEEE, 2020.

[69] Davide Mambelli, Frederik Träuble, Stefan Bauer, Bernhard Schölkopf, and Francesco Locatello. Compositional multi-object reinforcement learning with linear relation networks. In *ICLR2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*, 2022. URL https://openreview.net/forum?id=HFUxPr_I5ec.

[70] Allan Zhou, Vikash Kumar, Chelsea Finn, and Aravind Rajeswaran. Policy architectures for compositional generalization in control. In *ICML Workshop on Spurious Correlations, Invariance and Stability*, 2022.

[71] Michael Chang, Alyssa Li Dayan, Franziska Meier, Thomas L. Griffiths, Sergey Levine, and Amy Zhang. Hierarchical abstraction for combinatorial generalization in object rearrangement. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=fGG6vHp3W9W.

[72] Zhongwei Yu, Jingqing Ruan, and Dengpeng Xing. Learning causal dynamics models in object-oriented environments. In *International Conference on Machine Learning*, pages 57597–57638. PMLR, 2024.

[73] Caleb Chuck, Fan Feng, Carl Qi, Chang Shi, Siddhant Agarwal, Amy Zhang, and Scott Niekum. Null counterfactual factor interactions for goal-conditioned reinforcement learning. *arXiv preprint arXiv:2505.03172*, 2025.

[74] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2): 181–211, 1999.

[75] Shubham Pateria, Budhitama Subagdja, Ah-hwee Tan, and Chai Quek. Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 54(5):1–35, 2021.

[76] 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 (NeurIPS)*, 35:20064–20076, 2022.

[77] Nicholas Watters, Loic Matthey, Matko Bosnjak, Christopher P Burgess, and Alexander Lerchner. Cobra: Data-efficient model-based rl through unsupervised object discovery and curiosity-driven exploration. *arXiv preprint arXiv:1905.09275*, 2019.

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

[79] Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Elliott Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, Andrey Kurenkov, Karen Liu, Hyowon Gweon, Jiajun Wu, Li Fei-Fei, and Silvio Savarese. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 455–465. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/li22b.html.

[80] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2023.

[81] Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=Oxh5CstDJU.

[82] Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Elliott Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, Andrey Kurenkov, Karen Liu, Hyowon Gweon, Jiajun Wu, Li Fei-Fei, and Silvio Savarese. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, volume 164, pages 455–465. PMLR, 2022.

[83] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[84] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning (ICML)*, pages 2688–2697. PMLR, 2018.

[85] Colin Graber and Alexander G Schwing. Dynamic neural relational inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8513–8522, 2020.

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

[87] Silviu Pitis, Elliot Creager, Ajay Mandlekar, and Animesh Garg. Mocoda: Model-based counterfactual data augmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:18143–18156, 2022.

[88] Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, et al. Video language planning. *arXiv preprint arXiv:2310.10625*, 2023.

[89] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

[90] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

[91] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[92] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[93] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[94] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi\_\{0.5\}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[95] Yifan Zhong, Fengshuo Bai, Shaofei Cai, Xuchuan Huang, Zhang Chen, Xiaowei Zhang, Yuanfei Wang, Shaoyang Guo, Tianrui Guan, Ka Nam Lui, et al. A survey on vision-language-action models: An action tokenization perspective. *arXiv preprint arXiv:2507.01925*, 2025.

[96] Francesco Capuano, Caroline Pascal, Adil Zouitine, Thomas Wolf, and Michel Aractingi. Robot learning: A tutorial. *arXiv preprint arXiv:2510.12403*, 2025.

[97] Kento Kawaharazuka, Jihoon Oh, Jun Yamada, Ingmar Posner, and Yuke Zhu. Vision-language-action models for robotics: A review towards real-world applications. *IEEE Access*, 2025.

[98] Cheol-Hui Min, Jinseok Bae, Junho Lee, and Young Min Kim. Gatsbi: Generative agent-centric spatio-temporal object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3074–3083, 2021.

[99] Stefano Ferraro, Pietro Mazzaglia, Tim Verbelen, and Bart Dhoedt. Focus: Object-centric world models for robotic manipulation. *Frontiers in Neurorobotics*, 19:1585386, 2025.

[100] Avinash Kori, Ben Glocker, Bernhard Schölkopf, and Francesco Locatello. Unifying causal and object-centric representation learning allows causal composition. In *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*.

[101] Zhihong Deng, Jing Jiang, Guodong Long, and Chengqi Zhang. Causal reinforcement learning: A survey. *arXiv preprint arXiv:2307.01452*, 2023.

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

[103] Sumedh A Sontakke, Arash Mehrjou, Laurent Itti, and Bernhard Schölkopf. Causal curiosity: Rl agents discovering self-supervised experiments for causal representation learning. In *International conference on machine learning*, pages 9848–9858. PMLR, 2021.

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

[105] Hongye Cao, Fan Feng, Meng Fang, Shaokang Dong, Tianpei Yang, Jing Huo, and Yang Gao. Towards empowerment gain through causal structure learning in model-based rl. *arXiv preprint arXiv:2502.10077*, 2025.

[106] Zizhao Wang, Jiaheng Hu, Peter Stone, and Roberto Martín-Martín. Elden: exploration via local dependencies. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2023.

[107] Núria Armengol Urpí, Marco Bagatella, Marin Vlastelica, and Georg Martius. Causal action influence aware counterfactual data augmentation. In *Forty-first International Conference on Machine Learning (ICML)*, 2024.

[108] Haohong Lin, Wenhao Ding, Jian Chen, Laixi Shi, Jiacheng Zhu, Bo Li, and Ding Zhao. Because: Bilinear causal representation for generalizable offline model-based reinforcement learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[109] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

[110] Homanga Bharadhwaj, Kevin Xie, and Florian Shkurti. Model-predictive control via cross-entropy and gradient-based optimization. In *Learning for Dynamics and Control*, pages 277–286. PMLR, 2020.

[111] Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems*, 31, 2018.

[112] Phillip Lippe, Taco Cohen, and Efstratios Gavves. Efficient neural causal discovery without acyclicity constraints. *arXiv preprint arXiv:2107.10483*, 2021.

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: Yes, we verify the claims of our model in the empirical results on a set of simulated robotics and control tasks.

   Guidelines:

   - The answer NA 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 NA 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: We provide limitation discussions, especially on the usage of specific pre-trained models and the validation only on simulated benchmarks, but not real robots.

   Guidelines:

   - The answer NA 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: [NA]

Justification: Not a theoretical paper.

Guidelines:

- The answer NA 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: Yes, we will provide all essential details in the appendix.

   Guidelines:

   - The answer NA 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 will be publicly available after acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.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://nips.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, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Yes, we provide all details in the appendix, including the simulation setup, hyperparameters, and architecture design.

Guidelines:

- The answer NA 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:All runs are with either 5 or 10 random seeds (with error bars shown in the learning curve figures).

Guidelines:

- The answer NA 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]

   We provide the compute details in the appendix.

   Guidelines:

   - The answer NA 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: This research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA 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: [NA]

    Justification:This paper presents work at reinforcement learning and world models. While our research has potential societal implications, such as applications in robotics that could be misused, we do not identify any specific risks directly arising from our work that require explicit highlighting.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: There are no such risks.

    Guidelines:

    - The answer NA 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: Original methods are properly cited, and used environments, simulators, and tool are cited.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: No new assets.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: This research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA 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 (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

Appendix

A Overview

In this appendix, we provide supplementary details, extended discussions, and full results for FIOC-WM. Specifically, Section B presents an in-depth discussion of related work, including factored and object-centric reinforcement learning, hierarchical RL, and causality-inspired RL, all of which are relevant to our approach. Section C offers a detailed analysis of world model learning, focusing on the two-level factorization of state transitions (Section C.1), interaction modeling (Section C.2), and policy learning (Section C.3). Sections D, E, and F cover the experimental results, network architectures, and task specifications.

B Extended Related Works

B.1 Factored and Object-centric Reinforcement Learning

Our paper takes inspiration from multiple factorization frameworks in RL. Factored RL is based on the model of a Factored Markov Decision Process (MDP) [59], where structural relationships exist

among states, actions, and rewards. This factorization enables efficient policy solutions by leveraging these structural relationships [60, 61].

A specific type of factorization, which we also adopt, is object-based, referred to as object-centric reinforcement learning (object-centric RL) [62], where states or observations are grouped into object-centric clusters. In object-centric RL, actions typically target only a subset of objects, and rewards are often associated with the states of specific objects or object subsets. This object-wise factorization facilitates more structured and efficient decision-making.

Recent works on object-centric RL can be broadly categorized into two major directions: (i) learning object-centric representation and (ii) modeling the object relations and policy architectures for compositional generalization. For the first line, approaches focus on using object-centric representation learning techniques [48, 63–65] to extract meaningful object-level features from raw observations. These methods, based on object-centric representations, focus on policy learning that leverages the object-centric encodings to learn object-centric policies either directly from object-centric representations [66, 98, 22, 99]. Haramati et al. [67] further Consider the object interactions and learn the policy that has compositional generalization with the model-free framework, learning directly from images. The second line of work develops object-centric policies modeling object relations and policy structures, incorporating inductive biases in the state transition and policy networks. Methods include the use of graph neural networks [68], linear relational networks [69], self-attention, and deep sets [70], and then they leverage the modeled object-centric states and relational structures to achieve compositional generalization in reinforcement learning [71, 18, 33, 22, 73, 22]. Haramati et al. [67] further leverages object interactions to learn a policy with compositional generalization, using a model-free framework that learns directly from images. Our work combines ideas from both directions, which learn state factorization from the object-centric level (object-based factorization) and state level (dynamics and static factorization), and model the interactions among objects to achieve compositional generalization. Different from [22, 100], which explores a more fine-grained object-centric and attribute-level factorization, our approach demonstrates that a simpler dynamic-static factorization of objects is already sufficient for effective world model and policy learning, striking a balance between minimality and expressiveness. Additionally, we go beyond object-centric policies by learning an interaction-centric policy that leverages the learned interaction model to facilitate long-horizon task learning.

B.2 Hierarchical Reinforcement Learning

Our work, using low-level and high-level policy for decomposing complex tasks into interaction learning, is relevant to hierarchical reinforcement learning (HRL). HRL typically consists of a high-level policy (often referred to as the option framework [74], sub-skills, or sub-goals in the literature) and a low-level policy, enabling the efficient learning of complex RL tasks (see a recent survey [75]).

Within the scope of HRL, the most relevant to our work is the line of research that focuses on learning goal-conditioned hierarchical policies or hierarchical skill discovery. Within the scope of HRL, the closest to our work is the line of research that focuses on learning goal-conditioned hierarchical policies or hierarchical skill discovery. Zadaianchuk et al. [66] proposes the goal-conditioned hierarchical policies with learning object-based hierarchical goals. Hierarchical skill discovery focuses on decomposing complex tasks into object-wise or object-interaction-based components. For instance, Wang et al. [37] use conditional independence testing to identify sub-goals, while Chuck et al. [35] and [76] use Granger causality and causal models to uncover hierarchical structures. Our work shares similarities with [35, 37], particularly in using interactions as sub-skills. However, our work is on learning interaction models jointly with observations and dynamics within the world model directly from high-dimensional inputs, providing a more general and unified framework.

B.2.1 Causality-inspired Reinforcement Learning

Closely related to factored RL, causality-based RL aims to learn and leverage causal structures in Markov Decision Processes (MDPs) [101]. Building causal structures within MDPs or world models can enable more efficient exploitation[102–105] or policy learning [54, 106]. Additionally, learned causal structures can facilitate counterfactual reasoning, providing benefits such as counterfactual data augmentation to improve RL efficiency [86, 87, 107, 108]. Similarly, our work builds an interactive world model that aligns with this line of research, utilizing factored and causal structures in dynamic

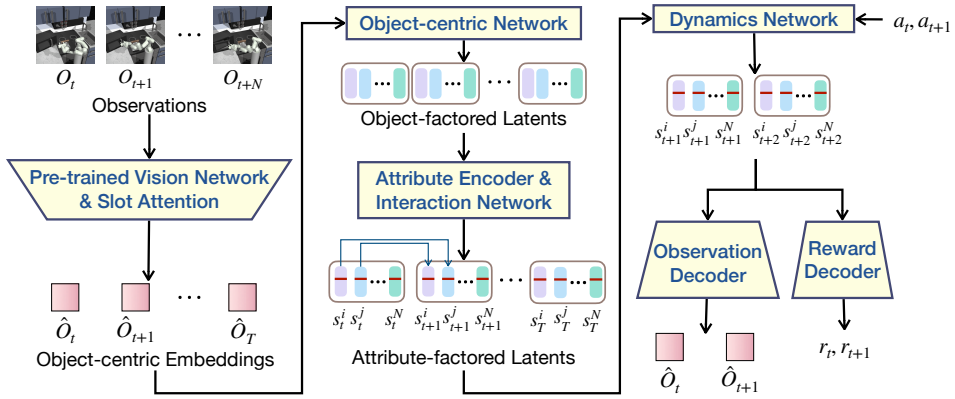

Figure A1: The pipeline of Offline Model Learning (Stage 1) jointly learns the observation function, state factorization, dynamics model, and reward model. Although Fig. 2 includes low-level policy learning as part of Stage 1, for clarity, we defer the discussion of low-level policy learning to Stage 2.

models to enhance generalization. Specifically, we focus on achieving compositional generalization at the levels of objects and their attributes.

The most relevant work to ours is SKILD [37], which also employs interaction-based hierarchical policies. However, there are several key differences. First, we aim to develop a general framework that incorporates state factorization, latent interaction-relevant states, and multiple approaches for learning interactions, including directly from pixel observations. In contrast, SKILD primarily focuses on state-based settings and learns interactions using conditional independence testing. Second, while SKILD is designed for unsupervised RL and skill discovery, our work focuses on general RL settings, although we consider extending it to unsupervised RL in future work. Despite these differences in scope and objectives, we acknowledge the contributions of SKILD, particularly in policy learning, and directly adopt certain components such as diversity measurement parameters and settings.

C Overall Framework

Fig. A1 illustrates the overall framework for learning FIOC-WN. In this section, we provide more supplementary details on this two-level factorization and interaction learning.

C.1 Graphical Representation of State Transitions

Fig.A2 provides an illustrative example that complements the graphical model in Fig.2 of the main paper, showing state transitions under dynamic graph structures (dashed red edges). We learn a two-level factorization of object-level and attribute-level representations, as detailed in Section 3.1. Here, we further motivate the choice of this two-level structure. First, decomposing a scene into individual objects reduces model complexity, as many objects move independently in most scenarios. By factoring dynamics and static attributes, the model can focus on learning the evolution of dynamic properties (e.g., position, velocity), while separately accounting for how static attributes (e.g., shape, mass) influence those dynamics. Second, to model interactions precisely and compactly, we focus on dynamic features being influenced by the attributes of interacting objects. For example, when two balls collide, it is primarily their dynamic attributes (e.g., velocity) that change, influenced by the full set of features (e.g., mass, shape, velocity) of the other object. This targeted interaction modeling allows the world model to be more precise and interpretable.

As a result, the learned world model benefits from this minimal yet sufficient factorization, which also enables more effective policy learning. The effectiveness of this design is further supported by ablation results shown in Table 3 in the main paper.

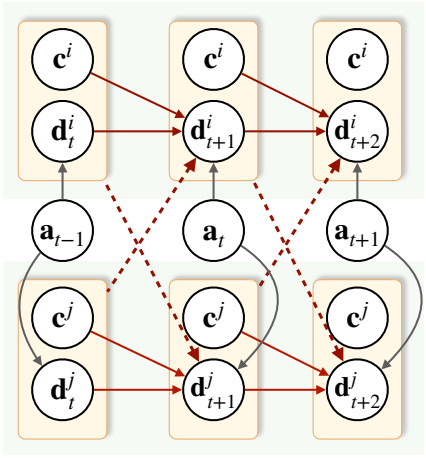

Figure A2: The detailed structure (example) of the state transitions with dynamic graph structure.

C.2 Learning the Interaction Models

C.2.1 Learning with Variational Masks

We consider two cases, where the masks are sampled from the approximated categorical distribution or from a codebook. In both cases, we have the ELBO [3]:

$$\mathcal{L}_{\text{mask}} = \mathbb{E}_{q_{\phi_u}(G|s)} \left[\log p_{\theta_u}(\mathbf{s} \mid G)\right] - \text{KL}\left[q_{\phi_u}(G \mid \mathbf{s}) \| p(G)\right] \tag{A1}$$

The encoders are parameterized by graph neural networks, following the settings in neural relational inference [84, 85] and amortized causal discovery frameworks [52]. $P(G)$ is the prior distribution of the graph structures. Specifically, for the encoder f_{enc}, ϕ_u, for each pair of nodes i and j, we compute:

$$\mathbf{u}_{ij} = f_{\text{enc},\phi_u}(\mathbf{s}^i, \mathbf{s}^j). \tag{A2}$$

For the categorical distribution one, we have:

$$G^{i,j} \sim \text{Softmax}\left(\mathbf{u}_{ij} + g/\tau\right), \tag{A3}$$

where τ is the temperature parameter and g is Gumbel-distributed noise [109].

For the latent codebook, we consider u as the learned latent embedding, while maintaining a codebook prototype vector set $\mathbf{e} = \{\mathbf{e}_1, \ldots, \mathbf{e}_k\}$, where k represents the number of different graph types (which can be interpreted as interaction patterns). Following [53], we apply a quantization step to discretize u:

$$e = e_z, \quad \text{where } z = \underset{j \in [k]}{\text{argmin}} \left\|\mathbf{u} - \mathbf{e}_j\right\|_2. \tag{A4}$$

After quantization, we retrieve the corresponding interaction graph by decoding the selected codebook entry into an adjacency matrix G:

$$G \sim g_{\text{dec}}(\mathbf{e}_z). \tag{A5}$$

All additional constraints and optimization techniques are directly adopted from [53].

[3] For clarity, in this section, we omit state factorization indices, as well as object and time indices, whenever they are not essential for computation or when they generally apply to factored variables across all objects and time steps.

C.2.2 Learning with Conditional Independence Testing

Following [54], we use conditional independence testing to identify object interactions. Specifically, for each pair of objects i and j, we compute the conditional mutual information (CMI) of the parameterized dynamics at each time step t:

$$\mathrm{CMI}_{i,j} = \mathop{\mathbb{E}}_{\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}^j_{t+1}} \left[\log \frac{p_s\left(\mathbf{s}^j_{t+1} \mid \mathbf{a}_t, \mathbf{s}_t\right)}{p_s\left(\mathbf{s}^j_{t+1} \mid \{\mathbf{a}_t, \mathbf{s}_t \backslash \mathbf{s}^i_t\}\right)} \right]$$

where p_s represents the state transition dynamics used in our framework. During testing, we use the same parameterized transition model. Instead, after computing CMI, G is encoded as the adjacent matrix in $\mathbb{R}^{N \times N}$, directly determining the interaction structure G. Since we employ an object-wise factorization, the number of required CMI tests remains manageable.

C.3 Online Policy Learning

In policy learning tasks, we only use the variational masks for learning the regime variables.

Low-level Policy For the low-level policy that invokes interactions, we consider two approaches: model predictive control (MPC) and RL policies.

At time step t, we are given the target interaction graph at $t + k$, denoted as $G^k_t = G^*$. Given the learned transition model P_u, we use s^i and $\mathbf{u}^g$ to infer the target states s^i_g and s^j_g. Using these inferred target states, we apply MPC to generate a sequence of actions that transitions the system from t to $t + k$ while minimizing the discrepancy between the predicted and target states.

Following Bharadhwaj et al. [110], Zhou et al. [27], we employ the cross-entropy method (CEM) for optimization. Specifically, we minimize the mean squared error (MSE) between the predicted state $\hat{\mathbf{s}}_{t+k}$ and the target state $\mathbf{s}_g$, given an action sequence $\mathbf{a}_t, \ldots, \mathbf{a}_{t+k-1}$ and the learned transition dynamics p_s:

$$\mathcal{L}_{\text{MPC}} = \|\mathbf{s}_{t+k} - \mathbf{s}_g\|_2^2 . \tag{A6}$$

At each iteration, we sample a population of action sequences from a Gaussian distribution and use the MSE loss to update the mean and covariance of the distribution via stochastic gradient descent.

For RL policies, we train the policy $\pi^l(\mathbf{a}_t \mid \mathbf{s}_t, \mathbf{s}_g, \mathbf{u}_g)$ using PPO [57], similarly to the setting in [37].

High-level Policy For high-level policy, we use the diversity measurement, for each object i, we have this intrinsic motivation to sample the j that has not been interacted. Hence, for an object set $\mathcal{S} = \{\mathbf{s}^1, \mathbf{s}^2, \ldots, \mathbf{s}^N\}$, at each time step t, we fix one object s_i and sample a subset $\mathcal{S}_t \subseteq \mathcal{S}$ based on the diversity reward r_{div} introduced that prioritizes diversity and avoids already-interacted objects. We learn this policy via PPO, using the same way as those in SKILD [106] but with the additional task reward.

D Full Results

D.1 Full Results on World Modeling

Interaction Learning Table A1 gives the full results on the interaction learning. We use Structural Hamming Distance (SHD) to verify the effectiveness of capturing interactions. SHD is a standard metric widely adopted in the relational inference and causal discovery literature [111, 112, 53]. Results show that the variational method with a categorical distribution as the prior achieves the best performance, while the variational approach with codebook latents performs second-best across domains. Conditional independence testing (CIT) also yields comparable results to these two methods. In contrast, relying solely on attention-based mechanisms is not robust across all settings, indicating that directly inferring interactions from the attention matrix is insufficient. For precise interaction modeling, the relational inference modules are necessary.

Algorithm 1 FIOC-WM: Offline World Model and Online Policy Learning (Simplified)

Require: Offline dataset $\mathcal{D} = \{(o_t, a_t, r_t)\}$; pre-trained visual encoder p_{pre}; inference model q_ϕ; transition model p_s; reward model p_r; interaction model p_u; high-level policy π^h; interaction (low-level) policy π^ℓ

Ensure: Policies π^h, π^ℓ and world model components

Stage 1: Offline World-Model Learning
for all $(o_t, a_t, r_t) \in \mathcal{D}$ **do**
  $\hat{o}_t \leftarrow p_{\text{pre}}(o_t)$ ▷ Encode observation with pre-trained vision model
  $s_t^i \leftarrow q_\phi(s_t^i \mid \hat{o}_t)$ ▷ Infer object-centric latent state
  Factorize s_t^i into $s_t^i = (d_t^i,\ c^i)$ ▷ dynamics- and attribute-level factors
  Learn reward $p_r(r_t \mid s_t, a_t)$
  Learn transition $p_s(s_{t+1} \mid s_t, a_t, G_t)$
  Infer interaction graphs $G_t \sim p_u(G_t \mid s_t)$
  Train interaction policy $\pi^\ell(a_t \mid s_t, G_t^g)$
end for

Stage 2: Online Policy Learning
repeat ▷ For each environment rollout episode
  Observe environment steps; encode current state s_t using world model
  $G_t^g \sim \pi^h(G_t^g \mid s_t)$ ▷ Select goal/target interaction graph (object-centric subgoal)
  $a_t \sim \pi^\ell(a_t \mid s_t, G_t^g)$ ▷ Sample low-level action conditioned on interaction
  Execute a_t; observe r_t, o_{t+1}; update s_{t+1}
  Update policies π^h, π^ℓ with collected data
until episode terminates

	Envs	Variational (Cat.)	Variational (Code.)	CIT	Attention-based
Single-Task	3 objects	$0.09_{\pm 0.04}$	$\underline{0.08}_{\pm 0.03}$	$\mathbf{0.06}_{\pm \mathbf{0.02}}$	$0.12_{\pm 0.04}$
	5 objects	$\mathbf{0.12}_{\pm \mathbf{0.07}}$	$0.15_{\pm 0.09}$	$\underline{0.13}_{\pm 0.06}$	$0.20_{\pm 0.07}$
	7 objects	$\mathbf{0.16}_{\pm \mathbf{0.10}}$	$\underline{0.19}_{\pm 0.08}$	$0.21_{\pm 0.07}$	$0.29_{\pm 0.12}$
	9 objects	$\mathbf{0.27}_{\pm \mathbf{0.10}}$	$\underline{0.31}_{\pm 0.09}$	$0.35_{\pm 0.15}$	$0.41_{\pm 0.14}$
Attri. Gen.	3 objects	$\mathbf{0.08}_{\pm \mathbf{0.02}}$	$0.11_{\pm 0.05}$	$\underline{0.08}_{\pm 0.03}$	$0.15_{\pm 0.02}$
	5 objects	$\underline{0.14}_{\pm 0.06}$	$0.17_{\pm 0.11}$	$\mathbf{0.13}_{\pm \mathbf{0.04}}$	$0.22_{\pm 0.11}$
	7 objects	$\mathbf{0.17}_{\pm \mathbf{0.12}}$	$\underline{0.22}_{\pm 0.13}$	$0.25_{\pm 0.06}$	$0.34_{\pm 0.15}$
	9 objects	$\mathbf{0.30}_{\pm \mathbf{0.10}}$	$0.36_{\pm 0.14}$	$\underline{0.35}_{\pm 0.19}$	$0.50_{\pm 0.18}$
Comp. Gen.	4 objects	$\underline{0.12}_{\pm 0.06}$	$0.13_{\pm 0.04}$	$\mathbf{0.09}_{\pm \mathbf{0.03}}$	$0.15_{\pm 0.06}$
	6 objects	$\underline{0.19}_{\pm 0.09}$	$0.20_{\pm 0.10}$	$\mathbf{0.19}_{\pm \mathbf{0.07}}$	$0.29_{\pm 0.11}$
	8 objects	$\mathbf{0.23}_{\pm \mathbf{0.13}}$	$0.28_{\pm 0.12}$	$\underline{0.5}_{\pm 0.09}$	$0.35_{\pm 0.14}$
	10 objects	$\mathbf{0.32}_{\pm \mathbf{0.14}}$	$\underline{0.37}_{\pm 0.12}$	$0.39_{\pm 0.13}$	$0.48_{\pm 0.16}$

Table A1: Full results on the normalized SHD of FIOC with different interaction learning models in predicting the ground truth interactions in Sprites World Environment. The **bold** values indicate the best-performing method, and the underlined ones are the second-best.

Reconstruction Table A3 presents the complete LPIPS results across domains. LPIPS [83] (lower is better) evaluates the comparison between the generated frames with ground-truth observations at both the pixel and perceptual levels. Our FIOC model achieves the best performance in most cases, particularly in environments with rich interactions such as Sprites, Fetch, and Libero. In other cases where DINO-WM performs best, our model remains competitive, with LPIPS scores within 0.05 of DINO-WM.

D.2 Policy Learning

Table A2 gives the full results on single-task learning, attribute generalization, compositional generalization, and skill generalization in policy learning tasks. The learning curves of single task learning are given in Fig. A3. For some simpler tasks in these single-task learning scenarios, existing baselines, particularly EIT and TD-MPC2, can achieve strong performance, for example, on Fetch. However,

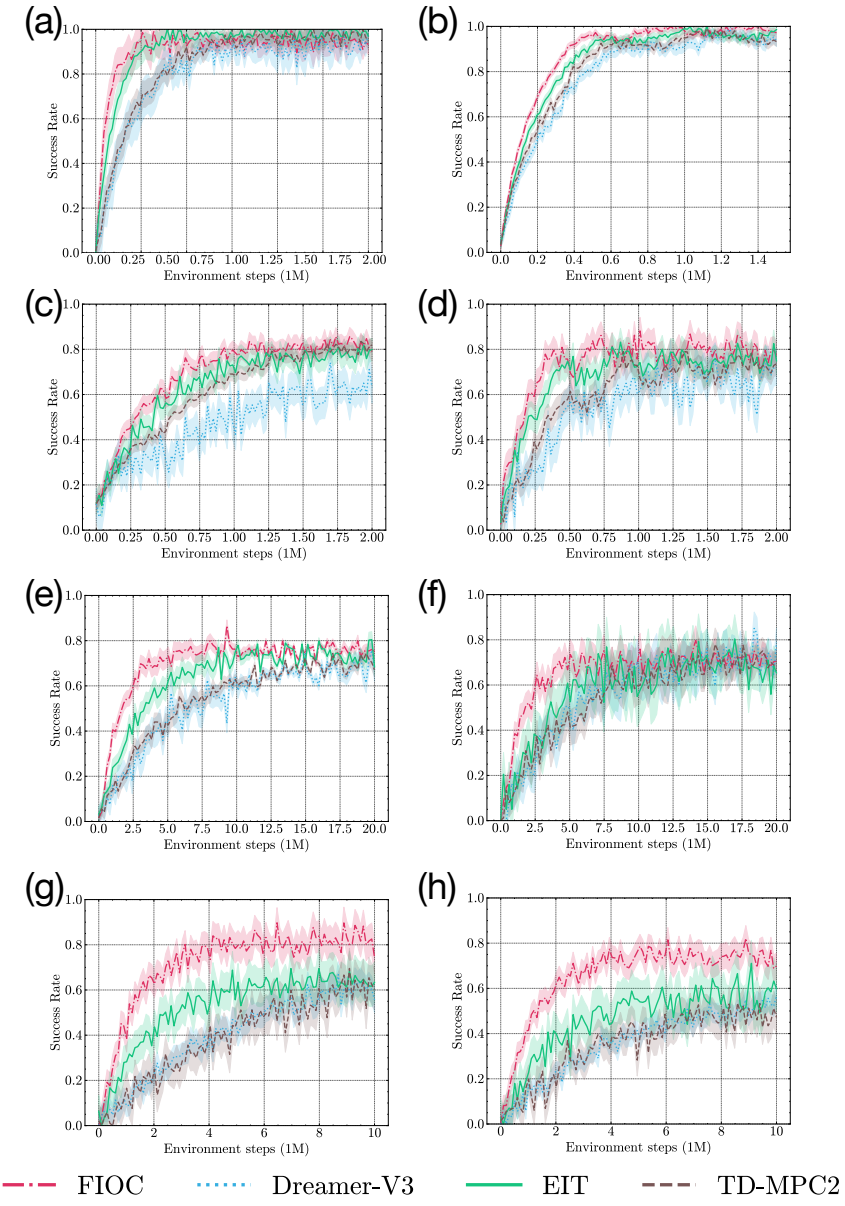

Figure A3: The policy learning curves of single task learning scenarios, (a). Gym Fetch Task 1; (b) Gym Fetch Task 2; (c) Franka Kitchen Task 1; (d) Franka Kitchen Task 2; (e). i-Gibson Task 1; (f). i-Gibson Task 2; (g). Libero Task 1; (h). Libero Task 2.

our method achieves the best performance in 5 out of 8 tasks, second-best in 1 task, and comparable performance (all $> 90\%$ success rate) on the remaining 2 tasks, which are relatively simple.

D.3 Full Ablations

Online World Model Fine-tuning For computational efficiency and because our experiments indicate that an offline-trained world model is already robust given high-quality offline data, we do not update the world model during the online stage by default. This is an empirical choice rather than a limitation; online updates are feasible. To assess the impact, we perform an ablation comparing performance with and without online world-model updates. Results are shown in Table A4.

	Envs	FIOC	Dreamer-V3	EIT	TD-MPC2
Single-Task	Gym Fetch (Task 1)	$0.95_{\pm 0.03}$	$\mathbf{0.98_{\pm 0.02}}$	$0.93_{\pm 0.05}$	$\underline{0.97}_{\pm 0.02}$
	Gym Fetch (Task 2)	$0.98_{\pm 0.01}$	$\mathbf{0.97_{\pm 0.02}}$	$0.95_{\pm 0.02}$	$\underline{0.96}_{\pm 0.02}$
	Franka Kitchen (Task 1)	$\underline{0.82}_{\pm 0.04}$	$0.75_{\pm 0.06}$	$0.69_{\pm 0.07}$	$\mathbf{0.83_{\pm 0.03}}$
	Franka Kitchen (Task 2)	$\underline{0.79}_{\pm 0.06}$	$0.68_{\pm 0.09}$	$0.75_{\pm 0.08}$	$\mathbf{0.73_{\pm 0.04}}$
	i-Gibson (Task 1)	$\mathbf{0.76_{\pm 0.12}}$	$0.69_{\pm 0.19}$	$\underline{0.74}_{\pm 0.14}$	$0.72_{\pm 0.12}$
	i-Gibson (Task 2)	$\mathbf{0.72_{\pm 0.10}}$	$0.68_{\pm 0.19}$	$\underline{0.74}_{\pm 0.14}$	$0.72_{\pm 0.12}$
	Libero (Task 1)	$\mathbf{0.81_{\pm 0.11}}$	$0.65_{\pm 0.14}$	$\underline{0.78}_{\pm 0.09}$	$0.76_{\pm 0.14}$
	Libero (Task 2)	$\mathbf{0.74_{\pm 0.09}}$	$0.58_{\pm 0.16}$	$\underline{0.69}_{\pm 0.07}$	$0.65_{\pm 0.12}$
Attri. Gen.	Push & Switch	$0.91_{\pm 0.05}$	$0.90_{\pm 0.07}$	$\underline{0.92}_{\pm 0.04}$	$\mathbf{0.95_{\pm 0.02}}$
	i-Gibson	$\mathbf{0.79_{\pm 0.13}}$	$0.62_{\pm 0.16}$	$\underline{0.70}_{\pm 0.14}$	$0.65_{\pm 0.15}$
	Libero	$\mathbf{0.76_{\pm 0.14}}$	$0.59_{\pm 0.18}$	$\underline{0.73}_{\pm 0.12}$	$0.69_{\pm 0.18}$
Comp. Gen.	Push & Switch	$\mathbf{0.86_{\pm 0.10}}$	$0.81_{\pm 0.12}$	$\underline{0.83}_{\pm 0.02}$	$0.79_{\pm 0.08}$
	Libero	$\mathbf{0.70_{\pm 0.09}}$	$0.58_{\pm 0.12}$	$\underline{0.65}_{\pm 0.08}$	$0.63_{\pm 0.14}$
Skill Gen.	Push & Switch	$\mathbf{0.81_{\pm 0.06}}$	$0.66_{\pm 0.10}$	$\underline{0.73}_{\pm 0.08}$	$0.65_{\pm 0.13}$
	Franka Kitchen	$\mathbf{0.73_{\pm 0.06}}$	$0.59_{\pm 0.09}$	$\underline{0.65}_{\pm 0.18}$	$0.62_{\pm 0.08}$

Table A2: Policy learning (success rate) of world model in Gym Fetch, Franka Kitchen, i-Gibson, and Libero tasks.

Environment	Dreamer-V3	TD-MPC2	EIT	DINO-WM	FIOC
Sprites	0.026	0.019	0.006	0.012	**0.004**
Fetch	0.042	0.039	0.026	0.009	**0.007**
Kitchen	0.102	0.123	0.096	**0.035**	0.038
i-Gbison	0.135	0.092	0.085	**0.063**	0.068
Libero	0.089	0.061	0.040	0.035	**0.027**

Table A3: Comparison of world models on LPIPS metrics.

Using pre-trained embeddings for Dreamer and TD-MPC Specifically, we adapted both baselines to use DINO embeddings as input:

- For Dreamer-V3, we replaced the pixel encoder with a frozen DINO encoder and used DINO-WM's decoder to reconstruct the observations.
- For TD-MPC2, we used DINO features to predict actions, terminal values, and rewards via its original decoding heads.

The updated results are shown in Table A5. The results indicate that the pre-trained DINO features improves both Dreamer-V3 and TD-MPC2 performance in some cases, but they still under-perform compared to FIOC. This suggests that while strong visual representations help, our proposed factorization (Stage 1) and policy learning (Stage 2) are key contributors to the performance gain.

Generalize to more objects To assess the model's ability to generalize to a greater number of objects, we train FIOC on environments containing three objects and evaluate on tasks with six and eight objects while keeping the world model fixed in Fetch Env. As shown in Table A6, FIOC achieves strong generalization performance, comparable to or better than baselines such as EIT, and it consistently outperforms Dreamer-V3 and TD-MPC2 under distribution shifts in object count.

Visualization Rollouts Some visualization rollouts are found at the project homepage: https://sites.google.com/view/fioc-wm.

Envs	FIOC (w/o Online Tuning)	FIOC (w/ Online Tuning)
Gym Fetch (Task 1)	0.95 ± 0.03	0.96 ± 0.02
Gym Fetch (Task 2)	0.98 ± 0.01	0.98 ± 0.01
Franka Kitchen (Task 1)	0.82 ± 0.04	0.79 ± 0.06
Franka Kitchen (Task 2)	0.79 ± 0.06	0.82 ± 0.05
i-Gibson (Task 1)	0.76 ± 0.12	0.78 ± 0.14
i-Gibson (Task 2)	0.72 ± 0.10	0.75 ± 0.06
Libero (Task 1)	0.81 ± 0.11	0.83 ± 0.08
Libero (Task 2)	0.74 ± 0.09	0.71 ± 0.06
Push & Switch (Attri. Gen.)	0.91 ± 0.05	0.95 ± 0.08
i-Gibson (Attri. Gen.)	0.79 ± 0.13	0.81 ± 0.15
Libero (Attri. Gen.)	0.76 ± 0.14	0.68 ± 0.09
Push & Switch (Comp. Gen.)	0.86 ± 0.10	0.82 ± 0.12
Libero (Comp. Gen.)	0.70 ± 0.09	0.74 ± 0.08
Push & Switch (Skill Gen.)	0.81 ± 0.06	0.82 ± 0.08
Franka Kitchen (Skill Gen.)	0.73 ± 0.06	0.72 ± 0.07

Table A4: Performance comparison of FIOC with and without online world model tuning across different environments. Values are mean $\pm$ standard deviation.

Envs	FIOC	Dreamer-V3 (DINO/original)	TD-MPC2 (DINO/original)
Kitchen	0.82	0.77 / 0.75	0.79 / 0.83
i-Gibson	0.76	0.71 / 0.69	0.73 / 0.72
Libero	0.81	0.69 / 0.65	0.74 / 0.76

Table A5: Comparison of FIOC with baselines using DINO features as input. Values indicate task success rates.

E Network Architectures and Hyper-parameters

E.1 World Models

Learning the Observation Functions For DINO-v2, we use ViT-Base for all cases. For R3M, we use ResNet-50 as backbones for all cases. For slot attention parameters, all settings remain consistent. Following VideoSAUR [47], we transform the original features using a two-layer MLP with an output dimension equal to the slot dimension. The slot attention module is initialized with randomly sampled slots to group the first-frame features. For subsequent frames, we initialize the slot attention module with the slots from the previous frame, which are additionally transformed using a predictor module with a GRU recurrent unit in the slot attention grouping.

For the VAE used to learn latent states, we employ a two-layer MLP with a hidden size of 256. The specific hyperparameters for different environments are detailed in Table A7.

Learning the Regime Variables For variational masks with a categorical distribution, we directly adopt the hyperparameters and network design from ACD [52]. However, unlike ACD, we use a GRU as the encoder, where the MLP has a hidden size of 256 with 3 layers. For the codebook-based approach, we use an MLP with 3 layers. The hidden layer size is set to 128 for Sprites-World, while for other environments, it is 256. The number of the centered codes are 16 for Sprites World, 8 for Gym Fetch, and 10 for others. All training hyperparameters follow those specified in [52] and [53].

For conditional independence testing, the threshold hyperparameters are set as follows: 0.02 for Sprites-World, 0.15 for Gym-Fetch, and 0.05 for other environments. All remaining hyperparameters are shared across environments.

Learning the State Transitions The state transitions are modeled using MLP layers with different configurations across environments. Specifically, we use a 2-layer MLP with hidden dimensionality 32 and SiLU activation for Sprites-World, a 2-layer MLP with hidden dimensionality 64 for Gym-

Envs	FIOC (Ours)	Dreamer-V3	EIT	TD-MPC2
3 objects	0.93	0.96	0.94	**0.97**
6 objects	**0.81**	0.54	0.77	0.62
8 objects	**0.70**	0.44	0.62	0.53

Table A6: Generalization to increased object count. Models are trained with 3 objects and evaluated with 6 and 8 objects using a fixed world model. FIOC shows strong generalization, matching or exceeding EIT and outperforming Dreamer-V3 and TD-MPC2 under distribution shifts in object count.

Parameter	Values (shared if not specified)
Used VIT for DINO	Base
Used ResNet for R3M	ResNet-50
Patch Size	16
Feature Dimension	768
Gradient Norm Clip	0.05
Image Crop/Resize	64 (SpritesWorld), 224 (others)
Slots	Number of objects + 2
Iterations for Clustering	3
Slot Dimension	32 (SpritesWorld), 64 (Gym-Fetch), 128 (others)
Latent Dimensions for s^s, s^c	8, 6 (SpritesWorld); 10, 8 (Gym-Fetch); 16, 12 (others)

Table A7: Hyperparameters used in learning observation functions.

Fetch, and a 3-layer MLP with hidden dimensionality 128 for other environments. The one-step prediction GRU layer consists of 3 MLP layers, with hidden dimensionality 128 for Sprites-World and 256 for i-Gibson, Gym-Fetch, Libero, and Franka Kitchen. The detailed settings are provided in Table A8. All with the learning rate $3e-4$.

Environment	MLP Layers (State Transition)	GRU Layers (One-Step Prediction)
Sprites-World	2 layers, 32 hidden	3 layers, 128 hidden
Gym-Fetch	2 layers, 64 hidden	3 layers, 256 hidden
i-Gibson	3 layers, 128 hidden	3 layers, 256 hidden
Libero	3 layers, 128 hidden	3 layers, 256 hidden
Franka Kitchen	3 layers, 128 hidden	3 layers, 256 hidden

Table A8: Hyperparameters for state transition modeling across different environments.

Offline RL For offline RL experiments, we use the same hyper-parameters as the online ones. Same as DINO-WM [27], we do not use expert demonstrations and inverse dynamics models to learn the mapping $p(\mathbf{a}_t|\mathbf{s}_t, \mathbf{s}_{t+1})$.

Others For offline training, we collect 3000 episodes with random actions for Sprites-World. For all other environments, we collect 2000 episodes using pre-trained policies from Dreamer-v3. The hyperparameters for the loss terms are set as $\{\alpha, \beta, \gamma, \eta\} = \{1, 0.05, 0.1, 0.2\}$, and the learning rate is set to 3×10^{-4}. The detailed data collection settings are provided in Table A9.

E.2 Policy Learning

Low-Level Policy For MPC, we use gradient descent with a learning rate of 5×10^{-5}. For those using PPO, we set the learning rate to 3×10^{-4} with a clip ratio of 0.1. The MLP architecture consists of hidden sizes $[256, 256]$ for Gym-Fetch, while for other environments, we use $[512, 512]$. Generalized Advantage Estimation (GAE) is set to 0.95 for all environments, and the entropy coefficient is 0.1.

Environment	Number of Episodes	Action Strategy
Sprites-World	3000	Random Actions
Gym-Fetch, i-Gibson, Libero, Franka Kitchen	2000	Pre-trained Policies (Dreamer-v3)

Table A9: Offline training settings across different environments.

High-Level Policy For high-level policy learning, we use PPO with a learning rate of 1×10^{-4}. The MLP architecture follows the same structure as the low-level policy, with hidden sizes of $[256, 256]$ for Gym-Fetch and $[512, 512]$ for other environments. The batch size is 1024 for all.

E.3 Computes and Training Time

Compute used for training the FIOC-WM:

- For Sprites-World, we use 3 hours on 1x NVIDIA A100;
- For Fetch, we use 8 hours on 6x NVIDIA 4090;
- For i-Gibson, we use 9 hours on 6x NVIDIA 4090;
- For Libero-object, we use 8 hours on 1x NVIDIA A100;
- For Kitchen, we use 6 hours on 1x NVIDIA A100.

F Task Details

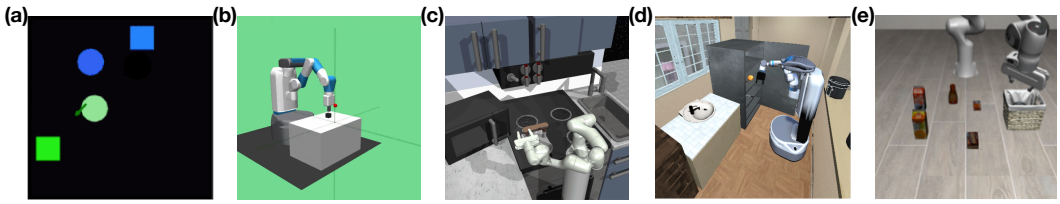

Figure A4: visualization of used benchmarks. From left to right: (a). Sprites-World; (b). OpenAI-Gym Fetch; (c). Franka Kitchen; (d). i-Gibson; and (e). Libero.

OpenAI Gym Fetch [78] is an environment featuring a Fetch robotic arm capable of manipulating cubes and switches. The tasks involve completing sub-tasks that require pushing or switching a varying number of objects. For single-task learning, we consider 2-push (Task 1) and 2-switch (Task 2), each with 2 million and 1.5 million training steps. For the attribute generalization task, we consider changing the color of the objects. For the compositional generalization task, we add one object to the push task, making it becoming the 3-push task. For skill one, we consider training both 2-push and 2-switch and compose them together for 2-push + 3-switch. All generalization tasks are evaluated with zero-shot generalization (for the skill generalization, we let the agent know the compositional task structure by providing separate rewards).

Franka-kitchen [40] is the environment where the 7-DoF Franka Emika Panda arm needs to perform tasks in the kitchen setup. Here we consider several sequential sub-tasks, such as turning on the microwave, moving the kettle, turning on the stove, and turning on the light. For single-task learning, we consider these two tasks:

- Task-1 is *Turn on the microwave - Move the kettle - Turn on the stove - Turn on the light*;
- Task-2 is *Turn on the microwave - Turn on the light - Slide the cabinet to the right - Open the cabinet*.

All tasks are with 2M training steps. The skill generalization one is *Turn on the microwave - Move the kettle - Slide the cabinet to the right - Open the cabinet*, evaluating with 0.2M training (10% as the base tasks).

i-Gibson [82] is a realistic environment with a simulated Fetch robot operating in everyday household tasks with rich objects and interactions. Similar to the setting in [37], we consider the tasks that

related to the peach object. The peach can be washed or cut, adding complexity to the tasks. The Task-1 is grasping the peach, Task-2 is cutting it with a knife. Each is with 20M steps to train. For attribute generalization, we change both the color and the size of the peach and follow the Task-1 setting with 1M adaptation steps.

Libero [80] is a benchmark designed for lifelong robot learning and imitation learning in household and tabletop environments. We focus on tasks randomly selected from the task library within libero-object. Task 1 and Task 2 involve picking two different sets of daily objects (boxes, cubes, and glasses) and moving them to a designated basket. The compositional generalization setting introduces objects with different colors and requires picking another randomly selected set of objects and placing them in the basket. The number of objects to manipulate is 5 for Task 1 and Task 2, while the generalization setting includes 7 objects. Number of training steps are 10M for the base tasks and 1M for the generalization task.