CoCoON: Contrastive Consistency Over Environments for Robotic Manipulation

Anonymous authors
Paper under double-blind review

Abstract

Robotic manipulation policies often generalize poorly to unseen environments. We study cross-environment generalization, where policies must generalize across heterogeneous environments with substantial variations in appearance, geometry, and spatial layout. Our key insight is task-state-conditioned environment invariance: visual representations should preserve task-relevant state information while suppressing environment-specific variations. Specifically, observations at similar manipulation states should have similar representations across environments. However, conventional contrastive learning is ill-suited to manipulation states that vary continuously rather than forming discrete instances or classes. We therefore construct continuous task-state similarities as soft contrastive supervision. This introduces two challenges: densely sampled trajectories produce diffuse and weakly discriminative soft targets, while similarity-based supervision does not explicitly align similar states across environments. We propose CoCoON (Contrastive Consistency Over Environments), a novel task-state-guided cross-environment contrastive learning framework. Concretely, we first construct continuous task-state similarities from task-relevant information as soft contrastive targets. Secondly, we introduce a Task-State Similarity Calibration module to sharpen the soft targets through stage gating and a centered-sigmoid transformation. Thirdly, we introduce Cross-Environment Feature Alignment to explicitly align features of similar manipulation states across environments. Extensive experiments across diverse manipulation tasks and heterogeneous environments demonstrate that CoCoON achieves state-of-the-art zero-shot generalization to unseen environments.

1 Introduction

Robotic manipulation is a fundamental capability for agents operating in complex environments, with applications ranging from industrial automation to household assistance. Vision is one of the richest sensing modalities for robotic perception in manipulation, providing dense, task-relevant information for the agent. Recent advances in deep reinforcement learning have enabled visuomotor policies to learn manipulation skills from visual observations (Kalashnikov et al., 2018; Laskin et al., 2020; Wang et al., 2021; Ze et al., 2023; Huang et al., 2025; Zhao et al., 2024). However, successfully deploying agents in novel environments remains an open challenge for robotic manipulation. Even minor differences between training and deployment environments, such as changes in illumination, background or surface textures, camera viewpoints, can lead to substantial degradation in policy performance (Batra & Sukhatme, 2025; Pang et al., 2025).

Visual robustness in robotic manipulation is commonly studied under domain-randomized variations, where predefined domain factors such as illumination, background or table textures, camera viewpoints, and distraction objects are varied (Tobin et al., 2017; James et al., 2019; Garcia et al., 2023; Pumacay et al., 2024). Such perturbations typically create visual shifts around a shared scene configuration. Nevertheless, robustness to these domain-randomized variations does not necessarily imply the generalization to novel scenes, where the geometry and spatial layout can differ substantially from those encountered during the training.

In this paper, we study cross-environment generalization for vision-based reinforcement learning in robotic manipulation. Unlike conventional domain randomization, which perturbs domain factors around a shared scene layout, our setting considers broader variations across different scene

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

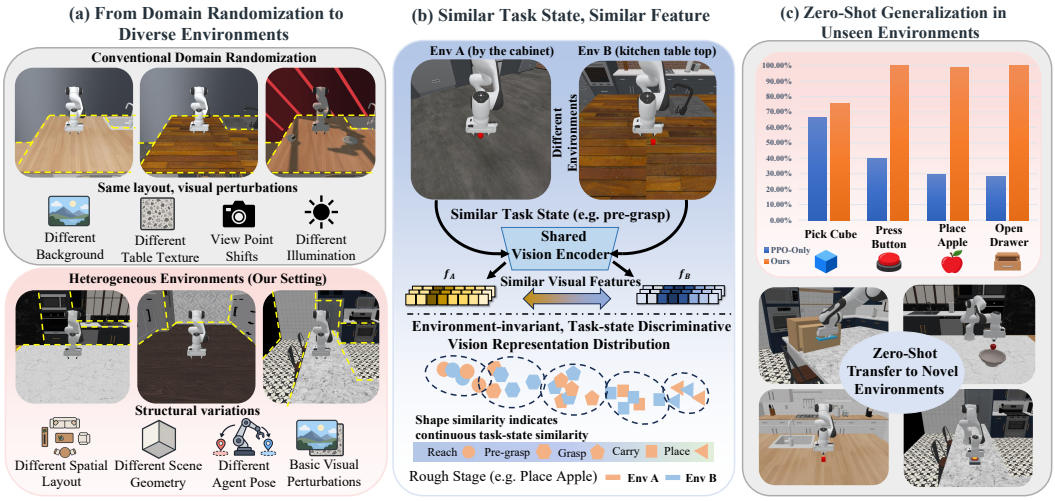


Figure 1: Motivation of CoCoON. (a) Beyond conventional domain randomization within a shared scene layout, we consider generalization across structurally diverse environments. (b) Our key insight is **task-state-conditioned environment invariance**: preserving task-state information while suppressing environment-specific variations. (c) Such representations enable zero-shot generalization to unseen environments.

configurations. Fig. 1 illustrates the distinction between conventional domain randomization and cross-environment generalization considered in our setting. For example, a policy trained on the kitchen floor may be deployed in a scene where the same task should be performed on a living room tabletop. Generalization across such heterogeneous environments can be challenging because visual reinforcement learning policies may overfit to environment-specific shortcuts during training (Song et al., 2020; Zhang et al., 2021; Stone et al., 2021; Dunion et al., 2023; Liu et al., 2023).

Our key insight is that environment invariance should not be enforced indiscriminately; instead, it should be conditioned on task-state similarity. Specifically, observations corresponding to similar manipulation states should be mapped to similar feature representations by the vision encoder, even when their surrounding environments differ in appearance, geometry, or spatial layout (Fig. 1(b)). This naturally motivates the usage of contrastive learning, which is a promising approach to encourage cross-environment representation invariance (Chen et al., 2020; Oord et al., 2018; He et al., 2020). Manipulation states vary continuously, making strict positive-negative assignments across different trajectories and environments inappropriate. In contrast to conventional contrastive learning methods, we aim to construct continuous pairwise similarities from task-relevant manipulation states for soft contrastive supervision. In addition, we identify two issues when applying this supervision. Firstly, dense trajectories contain many similar states, which can produce diffuse and weakly discriminative soft targets. Secondly, similarity-based supervision preserves relative state relationships, but does not explicitly align similar states between environments.

We propose CoCoON, a task-state-guided cross-environment contrastive learning framework that constructs continuous task-state similarities from task-relevant information, with calibrated soft targets and explicit feature alignment. We define a manipulation stage as a coarse phase of task execution determined by task-specific events, such as reaching, grasping, or transporting objects. Our pipeline consists of three stages. **1) Task-State Descriptor Construction.** We use task-relevant information obtained from the simulator to construct a descriptor of the current manipulation state. Manipulation stages are included in the descriptor. This enables the computation of continuous similarity between two samples. **2) State Similarity Modeling.** We treat observations from different stages as strict negatives and compute continuous similarities only within the same stage. We further calibrate these similarities with a centered sigmoid transformation to sharpen the soft target distribution. **3) Cross-Environment Feature Alignment.** We design an alignment loss to explicitly align the feature representations of observations across different environments. Intuitive effect of this alignment is shown in Fig 2(c). The learned vision encoder is incorporated into a PPO policy, which learns manipulation behaviors from the extracted visual features.

Our contribution can be summarized as follows:

- We introduce a new perspective on cross-environment generalization for robotic manipulation: *task-state-conditioned environment invariance*, which seeks to preserve task-relevant state information while suppressing environment-specific variations.
- We propose CoCoON, a task-state-guided cross-environment contrastive learning framework that constructs and calibrates continuous task-state similarities for soft contrastive supervision, with explicit feature alignment to enforce cross-environment consistency.
- Extensive experiments across diverse manipulation tasks and heterogeneous environments demonstrate that CoCoON consistently improves zero-shot generalization to unseen scenes, achieving state-of-the-art cross-environment generalization performance.

2 Related Work

Visual Generalization for Robotic Manipulation. Improving visual generalization has been an important problem in robotic manipulation. Early approaches tend to improve robustness of the learned policy by applying domain randomization and data augmentation (Tobin et al., 2017; James et al., 2019; Garcia et al., 2023), while recent work increasingly focuses on representation learning. Maniwhere and ReviWo learn view-invariant representation to improve robustness to camera viewpoint changes (Yuan et al., 2025; Pang et al., 2025). CAGE and ALDA explore task-relevant or disentangled representations under broader environmental changes (Xia et al., 2025; Batra & Sukhatme, 2025). Our work instead focuses on learning representations that remain invariant across environments while preserving continuous task-state information.

Contrastive Representation Learning. Contrastive learning has become a widely used paradigm for visual representation learning (Oord et al., 2018; Wu et al., 2018). Instance-discrimination methods such as SimCLR learn representations by treating augmented views of the same image as positives and other images as negatives (Chen et al., 2020), while momentum-based methods such as MoCo improve contrastive representation learning with large and consistent negative dictionaries (He et al., 2020). Supervised contrastive learning uses class labels to construct positive and negative pairs (Khosla et al., 2020). Contrastive objectives have also been used to learn transferable representations for downstream control and robotic tasks (Nair et al., 2023). More recent methods move beyond discrete labels. Rank-N-Contrast (RnC) incorporates continuous labels into contrastive representation learning by preserving their relative relationships, while RS-CL uses robot proprioceptive states to define similarities between observations for contrastive representation learning (Zha et al., 2023; Kim et al., 2026). Our work focuses on cross-environment robotic manipulation, where we construct pairwise similarities from task-relevant state information for soft supervision, while further addressing diffuse supervision and enforcing feature consistency across environments.

3 Method

Overview. Our overall goal is to enable a robotic manipulation policy to generalize reliably across diverse environments. To achieve this, we train a vision encoder to learn environment-invariant features. The overall pipeline is illustrated in Fig. 2. **Notation.** At each timestep t in environment e, the simulator returns an RGBD observation $o_t^{(e)}$. The vision encoder $E_\theta(o_t^{(e)})$ maps the observation to an embedding $f_t^{(e)} = E_\theta(o_t^{(e)})$. For a pair of observations o_i and o_j, we derive a task-state distance d_{ij} from task-relevant simulator information. We use $\tilde{s}_{ij}$ and $\tilde{q}_{ij}$ to denote the raw task-state similarity and its corresponding target distribution before calibration respectively. We use s_{ij} and q_{ij} to denote the calibrated task-state similarity and the final target distribution used for contrastive learning.

3.1 Task-State Similarity Modeling

We find it difficult to define positive and negative pairs of visual observations across different environments and trajectories, and grouping continuous manipulation states into discrete classes is inappropriate. Constructing meaningful similarity supervision between observations is also challenging. The simulators provide rich state information about the robot, manipulated objects, and

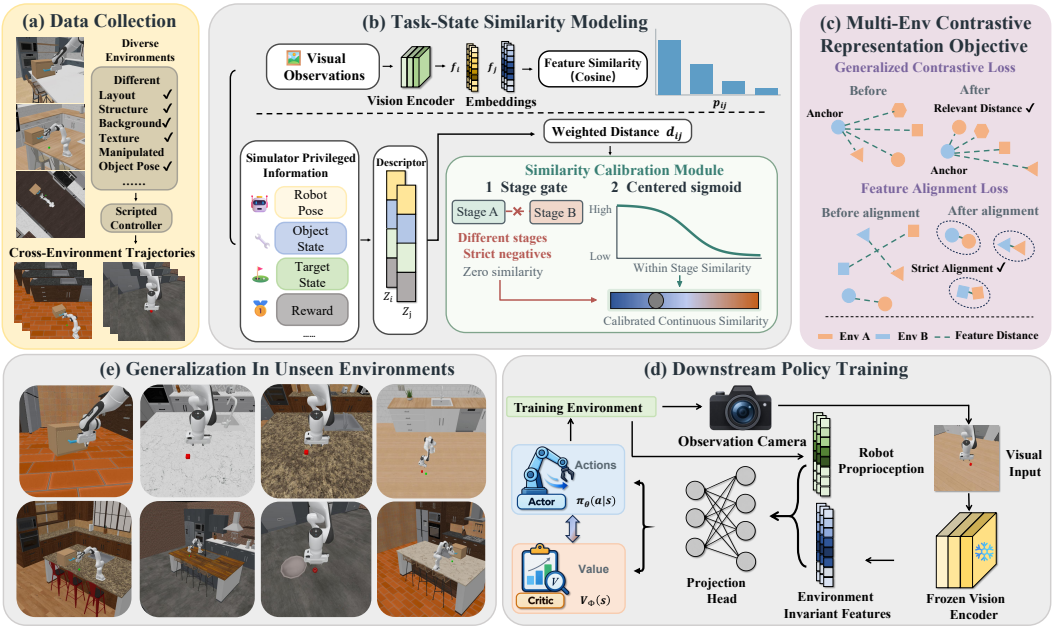


Figure 2: **The CoCoON pipeline.** (a) Cross-environment trajectories are collected in diverse environments via a scripted controller. (b) A task-state descriptor from simulator privileged information gives a weighted distance d_{ij}, calibrated into a discriminative continuous similarity through a stage gate and centered sigmoid. (c) The multi-environment objective couples a contrastive loss with a cross-environment feature alignment loss. (d) The frozen encoder feeds environment-invariant features to a PPO policy, (e) which generalizes to unseen environments.

progress of the task. Unlike environment appearance, which varies substantially across scenes, this information depends only on the underlying manipulation state and is independent of environment-specific visual variations. Therefore, we assign a task-state descriptor based on information provided by the simulator for each sample and compute the pairwise similarity between each sample using the descriptor.

Specifically, for each observation $o_t^{(e)}$, we construct the state descriptor $Z_{o_t^{(e)}}$ using information such as manipulated-object pose, target pose, reward, and interaction signals. The formulation of $Z_{o_t^{(e)}}$ varies between tasks, and the specific designs for each task are provided in Appendix B.2.

Given two observations o_i and o_j, we compute their task-state distance using the components of the two descriptors:

$$d_{ij} = \sum w_k \delta_k \left(Z_{i,k}, Z_{j,k} \right). \tag{1}$$

Here, $Z_{i,k}$ denotes the k-th component of Z_i, w_k is the corresponding weight, and δ_k denotes the normalized difference between the two components.

The resulting distance is converted into a continuous raw similarity as $\tilde{s}_{ij} = 1 - d_{ij}$, where $\tilde{s}_{ij} \in [0, 1]$ and a larger value indicates greater similarity between corresponding task states.

3.2 Cross-Environment Contrastive Learning

Consequently, conventional contrastive formulations are not well suited to the continuous state similarity since they rely on discrete positive and negative labels. We therefore use the pairwise task-state similarities as soft supervision across environments.

For each anchor observation o_i, we consider observations from different environments as candidate samples,

$$\mathcal{C}_i = \{ j \mid j \neq i,\ e_j \neq e_i \}. \tag{2}$$

The raw similarities are normalized over the candidate set to form a continuous target distribution,

$$\tilde{q}_{ij} = \frac{\tilde{s}_{ij}}{\sum_{k \in \mathcal{C}_i} \tilde{s}_{ik}}, \qquad j \in \mathcal{C}_i. \tag{3}$$

$\tilde{q}_{ij}$ preserves the continuous relationships among manipulation states.

We denote the cosine similarity between f_i and f_j as r_{ij}, and convert it into the feature distribution p_{ij} using a temperature-scaled softmax. The contrastive loss is then defined as (Yang et al., 2023):

$$\mathcal{L}_{\text{con}} = -\frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{C}_i} q_{ij} \log p_{ij}, \tag{4}$$

where $\mathcal{A}$ denotes the set of anchors. In this way, observations with similar manipulation states from different environments are encouraged to have higher feature similarity.

Directly normalizing the raw similarities may yield diffuse supervision in densely sampled manipulation trajectories, where many candidates correspond to similar states and receive similar values of $\tilde{s_{ij}}$. To quantify this effect, we compute the entropy of the target distribution for each anchor o_i:

$$H(\tilde{q}_i) = -\sum_{j \in \mathcal{C}_i} \tilde{q}_{ij} \log \tilde{q}_{ij}. \tag{5}$$

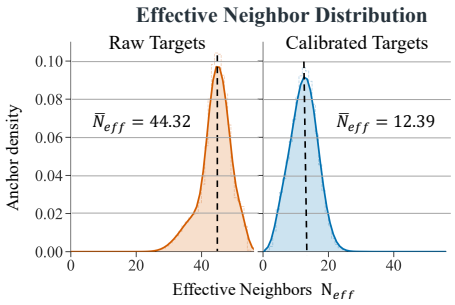


Figure 3: Effective neighbor distributions of the raw and calibrated targets, using OpenDrawer as an example. A large N_{eff} suggests that the target probability mass is effectively spread over many candidates. For a one-hot contrastive target with a single positive example, N_{eff} = 1.

Let $M_i = |\mathcal{C}_i|$ denote the number of valid cross-environment candidates. The entropy reaches its maximum value $\log M_i$ when all candidates receive equal target probability. We therefore normalize the entropy by this maximum:

$$H_{norm}(\tilde{q}_i) = \frac{H(\tilde{q}_i)}{logM_i}. \tag{6}$$

To provide a more intuitive interpretation of the target entropy, we further compute the effective number of candidates:

$$N_{eff}(\tilde{q}_i) = exp(H(\tilde{q}_i)) \tag{7}$$

N_{eff} can be interpreted as the number of equally weighted candidates that would yield the same entropy (Jost, 2006).

Together, H_{norm} and N_{eff} characterize the diffuseness of the contrastive target. As shown in 4.3, directly applying the generalized contrastive formulation results in a high normalized target entropy and a large number of effective candidates, which produces overly soft targets.

To address this issue, we reshape the similarity distribution. Specifically, we treat observation pairs from different stages (e.g., reaching, lifting, pushing, and carrying) as strict negatives while preserving continuous similarity among observations within the same stage. For within-stage pairs, we sharpen the similarity using a centered sigmoid transformation, which increases the separation between high and low similarity pairs while preserving their relative ordering.

$$s_{ij} = \begin{cases} \sigma(\alpha(c - d_{ij})), & g_i = g_j, \\ 0, & g_i \neq g_j, \end{cases} \tag{8}$$

where g_i denotes the manipulation stage, c is the distance center, and α controls the sharpness of the transformation. The same normalization in Eq. 3 is applied to s_{ij} to obtain q_{ij}.

3.3 Cross-Environment Feature Alignment

Although the contrastive loss encourages the learned features to preserve the relative similarity relationships among cross-environment samples, it does not explicitly constrain the absolute proximity of features corresponding to similar states. To further reduce the feature discrepancy between such samples, we introduce a feature alignment loss. For each valid cross-environment pair, we weight the alignment according to its similarity s_{ij}, so that pairs with higher similarity receive stronger alignment.

$$\mathcal{L}_{align} = \frac{\sum_i \sum_{j \in \mathcal{C}_i} s_{ij}(1 - r_{ij})}{\sum_i \sum_{j \in \mathcal{C}_i} s_{ij}} \tag{9}$$

This loss directly minimizes the cosine distance between features. Pairs with larger s_{ij} contribute more strongly to the alignment, while strict negatives have $s_{ij} = 0$ and therefore do not contribute to this term. We examine the effect of this alignment loss in 4.3.

3.4 Vision Encoder and Policy Training

We first train the vision encoder using trajectories collected from 100 diverse environments(Nasiriany et al., 2024). We collect 400,000 transitions for each task. The trajectories are generated by a scripted controller which uses privileged simulator states. No human teleoperation is required. The collected transitions are used to optimize the vision encoder with the representation objective

$$\mathcal{L}_{rep} = \mathcal{L}_{con} + \lambda \mathcal{L}_{align}. \tag{10}$$

We then freeze the vision encoder and use its output as visual input to the downstream reinforcement learning policy.

For downstream policy training, we follow the ManiSkill setup (Tao et al., 2025). At each timestep t, the visual observation o_t is encoded as $f_t = E_\theta(o_t)$. The visual feature f_t is concatenated with proprioceptive state and passes through a projection head. The projected representation is then provided to the actor and critic networks for PPO training(Schulman et al., 2017). The proprioceptive states only include internal measurement of the robot and are available at deployment. The agent is trained in a single environment and tested in multiple novel heterogeneous environments.

4 Experiments

4.1 Experimental Settings

Simulation Environments. We evaluate our method on four manipulation tasks using **ManiSkill3**(Tao et al., 2025), a robotic manipulation and learning framework built on the **SAPIEN** simulator (Xiang et al., 2020), and use RoboCasa (Nasiriany et al., 2024) to construct structurally diverse environments. For each room layout in RoboCasa, we further vary the position of the Franka robot to create different scene configurations. In total we construct 120 environments, including 100 environments for representation pretraining and 20 held-out environments for zero-shot evaluation. One of the 100 pretraining environments is selected as the downstream policy training environment. The 20 evaluation environments are unseen both during representation pretraining and downstream policy training.

Implementation Details. We evaluate four manipulation tasks, including PickCube, OpenDrawer, PlaceApple, and PushButton, covering grasping, articulated-object manipulation, object placement, and contact-based manipulation. All evaluated tasks are adapted to a Franka-based visual reinforcement learning setting. We adjust the control space according to the difficulties of each task. PickCube and PushButton allow the policy to control all the arm joints together with the gripper. For PlaceApple, we use a reduced action space consisting of end-effector Cartesian delta position control and a gripper command, while keeping the end-effector orientation fixed. We follow the visual observation setting of ManiSkill, using two camera views with 128×128 observations and the same lightweight CNN backbone. In addition, the policy receives the robot proprioceptive state. To ensure a fair comparison, all evaluated baselines include proprioceptive inputs and two camera views. We evaluate each policy on 20 unseen heterogeneous environments, with 7 evaluation

episodes per environment. We mark a policy as failed if its training return remains persistently low and its success rate on the training environment stays at 0% during the late stage of training.

Baselines. We compare CoCoON with six baselines spanning two directions: contrastive representation learning methods, including SimCLR, SupCon, RnC, and RS-CL, and pretrained visual encoders for robotic manipulation, including R3M and HRP (Chen et al., 2020; Khosla et al., 2020; Zha et al., 2023; Kim et al., 2026; Nair et al., 2023; Srirama et al., 2024). Notably, RS-CL is also specifically designed for robotic manipulation.

All contrastive representation-learning methods are pretrained on the same sampled manipulation trajectory data, after which the vision encoder is frozen for downstream PPO training. We adapt each method to our robotic manipulation setting while preserving its original learning principle. For **SimCLR**, two independently augmented views of the same transition observation form a positive pair, while views from all other observations in the minibatch are treated as negatives. No task-state labels are used. For **RS-CL**, we follow its state-aware formulation and use robot proprioceptive states to construct soft contrastive targets, encouraging observations from different environments with similar robot states to produce similar feature representations. For **SupCon**, we perform k-means clustering over the state descriptor Z, and use the resulting cluster assignments as categorical class labels for supervised contrastive learning. This encourages cross-environment features to cluster by task state rather than by environment. For **RnC**, Z is used for continuous supervision, and the representation is trained to preserve the relative ordering of pairwise task-state similarities in the feature space.

For robotic manipulation baselines, we directly use their publicly released pretrained weights. Except for the vision encoder, all downstream policy architectures, inputs, training settings are kept identical to CoCoON for a fair comparison.

4.2 Cross-Environment Generalization Results

Table 1: Cross-environment generalization results (%) on four manipulation tasks. Success rates are evaluated on unseen environments. Failed policies are assigned 0% when computing the average.

Method	PickCube	OpenDrawer	PlaceApple	PushButton	Avg.	Δ over PPO
PPO-only	66.43	27.86	29.29	40.00	40.89	–
SimCLR(Chen et al., 2020)	0.00	36.43	Fail†	47.86	21.07	-19.82
SupCon(Khosla et al., 2020)	64.29	86.43	Fail†	35.71	46.60	+5.71
RnC(Zha et al., 2023)	27.14	96.43	85.71	79.29	72.14	+31.25
RS-CL(Kim et al., 2026)	29.29	99.29	74.29	57.14	65.00	+24.11
R3M(Nair et al., 2023)	0.00	3.57	0.00	30.71	8.57	-32.32
HRP(Srirama et al., 2024)	11.43	39.29	17.86	70.71	34.82	-6.07
CoCoON	**75.71**	**100.00**	**98.57**	**100.00**	**93.57**	**+52.68**

† Fail indicates that the downstream policy fails to acquire the task on the training environment. All pretrained encoders are frozen during policy training. Therefore, if a pretrained representation does not sufficiently capture task-relevant information, the downstream policy may fail to acquire the task. Failed runs are assigned 0% success when computing the average. Δ denotes the average success-rate difference from PPO-only in percentage points.

Table 1 reports the policy success rates on 20 unseen environments across four manipulation tasks. CoCoON consistently achieves strong cross-environment generalization across all tasks, obtaining an average success rate of 93.57%.

SimCLR only treats two augmented views of the same observation as positives, while observation from different time steps are treated as negatives, ignoring the continuous task-state relationships along a manipulation trajectory. It relies on predefined augmentations that cannot fully cover cross-environment observation variations. SupCon groups features according to discrete task-state classes but does not constrain the relative distances between different classes. RnC only constrains the relative ordering of feature similarities, but task-equivalent observations may still remain far apart

Table 2: Comparison of target entropy and effective neighbors before and after target calibration. A normalized entropy close to 1 indicates that many candidates receive comparable supervision and are weakly distinguished by the target.

	Calibrated Target		Raw Target	
Task	**Entropy**	$\mathbf{N}_{\text{eff}}$	**Entropy**	$\mathbf{N}_{\text{eff}}$
PickCube	0.6985	18.26	0.9746	50.64
OpenDrawer	0.6095	12.39	0.9404	44.32
PlaceApple	0.6046	13.09	0.9245	42.09
PushButton	0.6557	16.28	0.9461	45.49

in the feature space. RS-CL uses proprioceptive similarity as supervision, but similar task state can correspond to different robot configurations, making proprioceptive similarity an imperfect proxy for task equivalence. These limitations may be acceptable in their original application domains, but become critical in cross-environment robotic manipulation.

R3M and HRP perform poorly in our cross-environment setting. R3M focuses on temporal and language-aligned representations from human videos, while HRP emphasizes hand, object, and contact affordances. Both methods learn general manipulation-oriented representations from large-scale human visual data, but are not specifically designed for zero-shot generalization across heterogeneous environments.

CoCoON addresses these limitations by preserving continuous task-state structure while explicitly aligning task-equivalent observations across environments.

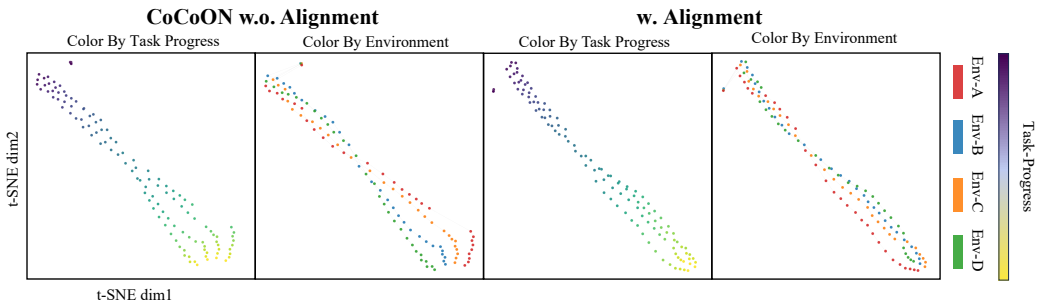


Figure 4: Identical trajectory replayed in four environments. t-SNE visualization of learned features with and without cross-environment alignment.

4.3 Ablation Studies and Analysis

Target Calibration Analysis. We examine how the target construction affects the supervision provided by contrastive learning by comparing the raw targets with calibrated targets on four manipulation tasks. Table 2 reports the normalized target entropy and effective number of candidates defined in Sec. 3.2. The raw targets exhibit high normalized entropy and large effective candidate counts, while the calibrated targets reduce both metrics across all four tasks. A high normalized entropy with a large effective candidate count reflects a diffuse target distribution that provides limited discriminative guidance for contrastive learning.

Feature Alignment Analysis. We examine the effect of the feature alignment mechanism on the learned representation. Figure 4 compares t-SNE visualizations of features learned with and without the alignment supervision. After introducing feature alignment, features corresponding to similar task states from different environments become noticeably more tightly aligned while preserving the overall task-state structure.

Overall Ablation. We evaluate the contribution of CoCoON by comparing a variant without contrastive supervision (PPO-only), a variant without target calibration and feature alignment, and a

variant without feature alignment with our full model. CoCoON achieves an average success rate of 93.57%, compared with 40.89% for PPO-only, demonstrating the overall benefit of the cross-environment contrastive learning method. Without calibration and alignment, the average success rate drops to 68.75%. Introducing target calibration increases the average success rate to 91.43%, adding feature alignment further improves the average to 93.57%. The results are shown in table 3.

Table 3: Ablation results on four manipulation tasks. Success rates (%) are evaluated on unseen environments. Failed policies are assigned 0 when computing the average.

Method	PickCube	OpenDrawer	PlaceApple	PushButton	Avg.	$\Delta \downarrow$
CoCoON	**75.71**	**100.00**	**98.57**	**100.00**	**93.57**	–
w.o. Alignment	71.43	100.00	94.29	100.00	91.43	2.14
w.o. Calibration & Alignment	78.57	96.43	Fail†	100.00	68.75	24.82
PPO-only	66.43	27.86	29.29	40.00	40.89	52.68

† Fail indicates that the downstream policy fails to acquire the task on the training environment. Failed runs are assigned 0% success when computing the average. $\Delta \downarrow$ denotes the decrease in average success rate from the full CoCoON model, in percentage points.

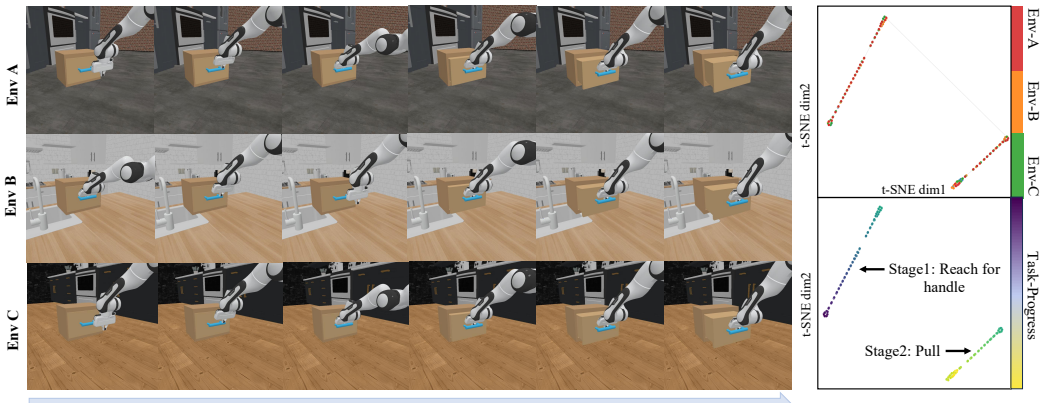


Figure 5: Representation Analysis.

Representation Analysis. We examine the structure of the visual representation learned by our vision encoder. Figure 5 visualizes the t-SNE embeddings of a same manipulation trajectory replayed in 3 environments. Although the observations exhibit substantial visual differences across scenes, features from the same step are located close to each other in the embedding space. Moreover, the features evolve continuously along the same manipulation stage and are clearly separated across different stages.

5 Conclusion.

We presented CoCoON, a contrastive representation learning framework for improving cross-environment generalization in visual reinforcement learning for robotic manipulation. CoCoON learns environment-invariant visual representations using task-relevant continuous contrastive supervision and explicit cross-environment feature alignment. Our experiments across multiple manipulation tasks demonstrate that CoCoON substantially improves policy generalization to unseen environments. Despite its strong cross-environment generalization, CoCoON still requires separate encoder pretraining for each task, and its generalization across object categories and robot embodiments remains unexplored. Future work will focus on generalization across different object instances within the same category.

AI USE STATEMENT

In this work, we used generative AI tools to assist with method implementation, identify potentially relevant work, and provide suggestions for improving the readability of the manuscript. All cited literature was read and verified by the authors, and AI-assisted writing suggestions were used only as editorial guidance. The authors take responsibility for the final content of this work, including the implementation, experimental results, claims, and manuscript.

REPRODUCIBILITY STATEMENT

We provide detailed task definitions, control settings, reward functions, PPO hyperparameters, task-state similarity construction, representation learning settings, and baseline implementation details in the Appendix. These details are intended to facilitate reproduction of our experiments. We will release our source code and trained model weights upon publication.

REFERENCES

David Arthur, Sergei Vassilvitskii, et al. k-means++: The advantages of careful seeding. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, volume 7, pp. 1027–1035, 2007.

Sumeet Batra and Gaurav S. Sukhatme. Zero shot generalization of vision-based RL without data augmentation. In *International Conference on Machine Learning (ICML)*, pp. 3251–3273. PMLR, 2025.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pp. 1597–1607. PMLR, 2020.

Mhairi Dunion, Trevor McInroe, Kevin Luck, Josiah Hanna, and Stefano Albrecht. Conditional mutual information for disentangled representations in reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:80111–80129, 2023.

Ricardo Garcia, Robin Strudel, Shizhe Chen, Etienne Arlaud, Ivan Laptev, and Cordelia Schmid. Robust visual sim-to-real transfer for robotic manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 992–999. IEEE, 2023.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9726–9735. IEEE, 2020.

Suning Huang, Zheyu Aqa Zhang, Tianhai Liang, Yihan Xu, Zhehao Kou, Chenhao Lu, Guowei Xu, Zhengrong Xue, and Huazhe Xu. MENTOR: Mixture-of-experts network with task-oriented perturbation for visual reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 26143–26161. PMLR, 2025.

Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12619–12629. IEEE, 2019.

Lou Jost. Entropy and diversity. *Oikos*, 113(2):363–375, 2006.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning (CoRL)*, pp. 651–673. PMLR, 2018.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:18661–18673, 2020.

Taeyoung Kim, Jimin Lee, Myungkyu Koo, Dongyoung Kim, Kyungmin Lee, Changyeon Kim, Younggyo Seo, and Jinwoo Shin. Contrastive representation regularization for vision-language-action models. In *International Conference on Machine Learning (ICML)*, 2026.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 5639–5650. PMLR, 2020.

Qiyuan Liu, Qi Zhou, Rui Yang, and Jie Wang. Robust representation learning by clustering with bisimulation metrics for visual reinforcement learning with distractions. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 37, pp. 8843–8851, 2023.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *Conference on Robot Learning (CoRL)*, pp. 892–909. PMLR, 2023.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024.

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

Jing-Cheng Pang, Nan Tang, Kaiyuan Li, Yuting Tang, Xin-Qiang Cai, Zhen-Yu Zhang, Gang Niu, Masashi Sugiyama, and Yang Yu. Learning view-invariant world models for visual robotic manipulation. In *International Conference on Learning Representations (ICLR)*, volume 2025, pp. 54853–54876, 2025.

Wilbert Pumacay, Ishika Singh, Jiafei Duan, Ranjay Krishna, Jesse Thomason, and Dieter Fox. The colosseum: A benchmark for evaluating generalization for robotic manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

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

Xingyou Song, Yiding Jiang, Stephen Tu, Yilun Du, and Behnam Neyshabur. Observational overfitting in reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Mohan Kumar Srirama, Sudeep Dasari, Shikhar Bahl, and Abhinav Gupta. Hrp: Human affordances for robotic pre-training. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

Austin Stone, Oscar Ramirez, Kurt Konolige, and Rico Jonschkowski. The distracting control suite–a challenging benchmark for reinforcement learning from pixels. *arXiv preprint arXiv:2101.02722*, 2021.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, et al. Demonstrating gpu parallelized robot simulation and rendering for generalizable embodied ai with maniskill3. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30. IEEE, 2017.

Yufei Wang, Narasimhan Gautham, Xingyu Lin, Brian Okorn, and David Held. Roll: Visual self-supervised reinforcement learning with object reasoning. In *Conference on Robot Learning (CoRL)*, pp. 1030–1048. PMLR, 2021.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3733–3742, 2018.

Shangning Xia, Hongjie Fang, Cewu Lu, and Hao-Shu Fang. Cage: Causal attention enables data-efficient generalizable robotic manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13242–13249. IEEE, 2025.

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11094–11104. IEEE, 2020.

Yuzhe Yang, Xin Liu, Jiang Wu, Silviu Borac, Dina Katabi, Ming-Zher Poh, and Daniel McDuff. Simper: Simple self-supervised learning of periodic targets. In *International Conference on Learning Representations (ICLR)*, 2023.

Zhecheng Yuan, Tianming Wei, Shuiqi Cheng, Gu Zhang, Yuanpei Chen, and Huazhe Xu. Learning to manipulate anywhere: A visual generalizable framework for reinforcement learning. In *Conference on Robot Learning (CoRL)*, pp. 1815–1833. PMLR, 2025.

Yanjie Ze, Yuyao Liu, Ruizhe Shi, Jiaxin Qin, Zhecheng Yuan, Jiashun Wang, and Huazhe Xu. H-index: Visual reinforcement learning with hand-informed representations for dexterous manipulation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 74394–74409. Curran Associates, Inc., 2023.

Kaiwen Zha, Peng Cao, Jeany Son, Yuzhe Yang, and Dina Katabi. Rank-n-contrast: Learning continuous representations for regression. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:17882–17903, 2023.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations (ICLR)*, 2021.

Ruihan Zhao, Sandeep P Chinchali, Mariano Phielipp, et al. Accelerating visual sparse-reward learning with latent nearest-demonstration-guided explorations. In *Conference on Robot Learning (CoRL)*, 2024.

APPENDIX

A TASK TRAINING DETAILS

A.1 TASK DEFINITIONS AND CONTROL SETTINGS

Table 4: Task definitions, control settings, and success criteria.

Task	Goal	Control Space	Success Criterion
PickCube	Grasp the cube and move it to a predefined virtual target position.	7-dimensional Franka joint-position delta control with a 1-dimensional gripper command.	Cube–target distance ≤ 2.5 cm and the robot is static.
OpenDrawer	Open a drawer with a handle.	7-dimensional Franka joint-position delta control.	Drawer opening ratio $> 75\%$ and linear velocity < 0.1 m/s.
PlaceApple	Pick up an apple and place it into a bowl.	End-effector Cartesian delta position control $(\Delta x, \Delta y, \Delta z)$ with a gripper command; end-effector orientation fixed downward.	Horizontal distance to the bowl center ≤ 5 cm, with the apple inside the bowl and released from the gripper.
PushButton	push the button to activate it.	7-dimensional Franka joint-position delta control with a 1-dimensional gripper command.	Button push ratio $\geq 75\%$; both gripper fingertips contact the button cap with ≥ 0.5 N contact force each.

A.2 HYPERPARAMETERS

Table 5: PPO training hyperparameters for the four manipulation tasks.

Hyperparameter	PickCube	OpenDrawer	PlaceApple	PushButton
Parallel environments	256	128	256	256
Episode horizon	50	100	50	50
Rollout steps	50	100	50	50
Samples per PPO rollout	12,800	12,800	12,800	12,800
Number of minibatches	8	32	8	8
Minibatch size	1,600	400	1,600	1,600
PPO update epochs	8	8	8	8
Total environment steps	10M	10M	10M	10M
Shared PPO Hyperparameters				
Learning rate	3×10^{-4}			
Discount factor γ	0.97			
GAE λ	0.9			
Advantage normalization	Enabled			
Clip coefficient	0.2			
Value coefficient	0.5			
Entropy coefficient	0			
Max gradient norm	0.5			
Target KL	0.2			

A.3 REWARD FUNCTIONS

The reward functions of four tasks are summarized in Table 6

Table 6: Dense reward functions for the four manipulation tasks.

Task	Reward Function
PickCube	$R_{\text{reach}} = 1 - \tanh(5d_{\text{tcp,cube}})$, $R_{\text{grasp}} = 1$, $R_{\text{goal}} = 1 - \tanh(5d_{\text{cube,goal}})$, $R_{\text{static}} = 1 - \tanh(5\|\dot{\mathbf{q}}\|_2)$, $R_{\text{success}} = 5$. R_{grasp} and R_{goal} are activated after grasping; R_{static} is activated after placement.
OpenDrawer	$R_{\text{reach}} = 1 - \tanh(5d_{\text{tcp,handle}})$, $R_{\text{open}} = 2\rho_{\text{drawer-open}}$, $R_{\text{reach}} = 2$ after opening starts, $R_{\text{open}} = 3$ at target opening, $R_{\text{success}} = 5$. $\rho_{\text{drawer-open}}$ denotes the drawer open ratio.
PlaceApple	$R_{\text{reach}} = 1 - \tanh(5d_{\text{tcp,apple}})$, $R_{\text{grasp}} = 1$, $R_{\text{lift}} = \lambda_{\text{lift}}\rho_{\text{lift}}$, $R_{\text{transport}} = 1 - \tanh(5d_{\text{apple,drop}})$, $R_{\text{release}} = 4 + R_{\text{transport}} + \rho_{\text{open}}$, $R_{\text{place}} = 6 + R_{\text{static}} + \rho_{\text{open}}$, $R_{\text{success}} = 80$. R_{lift} and $R_{\text{transport}}$ are activated after grasping.
PushButton	$R_{\text{reach}} = 1 - \tanh(5d_{\text{tcp,cap}})$, $R_{\text{reach}} = 1$ after pushing starts, $R_{\text{push}} = 2 + 4\rho_{\text{push}}$, $R_{\text{close}} = 0.2\rho_{\text{close}}$, $R_{\text{complete}} = 7 + R_{\text{static}}$.

Here, ρ_{lift}, ρ_{open}, ρ_{push}, and ρ_{close} denote normalized lifting, gripper-opening, button-pushing, and gripper-closing progress, respectively.

B Similarity Construction Details

B.1 Task-State Descriptors

Table 7 summarizes the task-state descriptors used for the four manipulation tasks.

Table 7: Task-state descriptors used for similarity computation.

Task	Dim.	Task-State Descriptor Z
PickCube	6	$[\Delta x, \Delta y, \Delta z$, contact, $d^{\text{gated}}_{\text{tcp,goal}}$, return-progress$]$
OpenDrawer	6	$[\Delta x, \Delta y, \Delta z$, hook, drawer-progress, return-progress$]$
PlaceApple	6	$[\Delta x, \Delta y, \Delta z$, is-grasped, apple-in-bowl, return-progress$]$
PushButton	7	$[\Delta x, \Delta y, \Delta z$, gripper-close, push-phase, push-fraction, return-progress$]$

B.2 Distance Computation

The distance between two observations is computed from the components of Z, with different weights used for different tasks and manipulation stages.

Table 8: Task-state distance definitions, normalization settings, and similarity calibration parameters.

Task	**Stage**	**XYZ Scale (m)**	**Distance d_{ij}**	**Sigmoid (c, α)**
PickCube	Pre-contact	$(0.08,\ 0.08,\ 0.15)$	$0.75\, d_{\text{geo}} + 0.25\, \Delta p$	$(0.30,\ 12)$
	Post-contact	–	$0.50\, \Delta d_{\text{goal}}^{\text{gated}} + 0.50\, \Delta p$	$(0.30,\ 12)$
PlaceApple	Pre-grasp	$(0.02,\ 0.02,\ 0.15)$	$0.75\, d_{\text{geo}} + 0.25\, \Delta p$	$(0.30,\ 12)$
	Carry	$(0.27,\ 0.27,\ 0.13)$	$0.75\, d_{\text{geo}} + 0.25\, \Delta p$	$(0.30,\ 12)$
OpenDrawer	Pre-hook	$(0.20,\ 0.05,\ 0.05)$	$0.75\, d_{\text{geo}} + 0.25\, \Delta p$	$(0.30,\ 12)$
	Post-hook	–	$1.00\, \Delta f_{\text{drawer}}$	$(0.30,\ 12)$
PushButton	Approach	$(0.08,\ 0.10,\ 0.08)$	$0.65\, d_{\text{geo}} + 0.15\, \Delta f_{\text{close}} + 0.20\, \Delta p$	$(0.25,\ 12)$
	pushing	–	$0.80\, \Delta f_{\text{push}} + 0.20\, \Delta p$	$(0.25,\ 12)$

C Representation Learning Details

Data Collection. We first train a PPO-only policy for each task and use its rollouts as references for constructing a scripted controller. With access to privileged simulator information, the controller generates trajectories that follow behavior patterns similar to those of the PPO policy. For each trajectory, we record the dual-view visual observations, robot proprioceptive states, rewards, and the final success outcome. Since reinforcement learning policies also encounter unsuccessful exploratory behaviors during training, we retain a small portion of failed trajectories. After filtering, approximately 80%-90% of the collected trajectories are successful.

Hyperparameters. The representation learning hyperparameters are summarized in Table 9

Table 9: Representation learning hyperparameters.

Hyperparameter	**Value**
Input	Dual-view RGB-D, 8 channels
Visual feature dimension	256
Feature temperature τ	0.1
Alignment weight λ_{align}	1.0
Optimizer	Adam, $\epsilon = 10^{-5}$
Learning rate	3×10^{-4}
LR schedule / warmup	None (fixed learning rate)
Batch size	64 (8 scenes $\times$ 8 frames)
Steps per epoch	1,000
Epoch	21

D Additional Representation Results

We further visualize the learned representations by replaying the same successful manipulation trajectory in four different environments for each task. Fig. 6 compares the t-SNE embeddings produced by PPO-only and CoCoON, with the same embeddings colored by environment or normalized task progress. The PPO-only features show strong environment-dependent separation, whereas CoCoON aligns features from different environments closely and preserves the continuous progression of the manipulation trajectory.

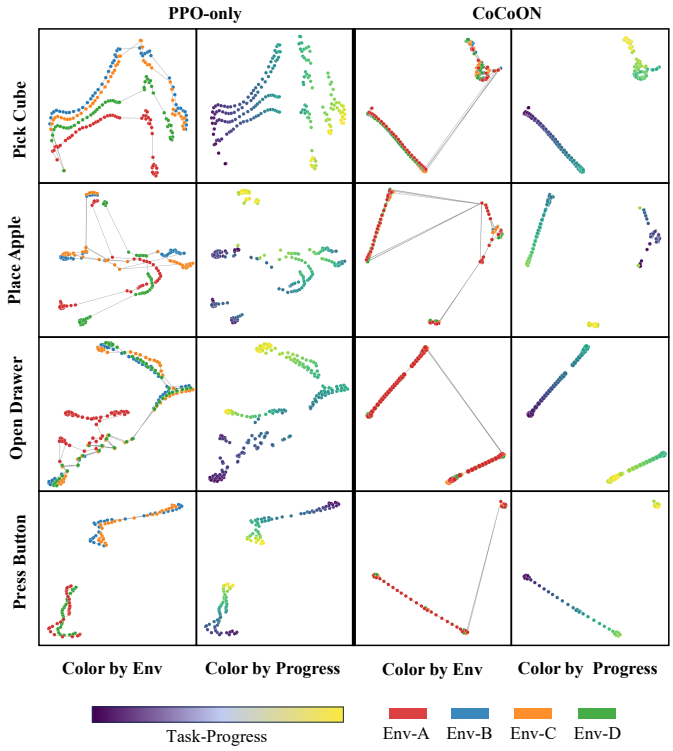


Figure 6: Each row corresponds to one task, and the columns show embeddings from PPO-only and CoCoON colored by environment and task progress.

E Baseline Implementation Details

All contrastive baselines use the same dual-view RGB-D input, visual backbone, pretraining data, and downstream PPO setup as CoCoON. After representation pretraining, the visual encoder is frozen for policy learning. We adapt each baseline to our setting while preserving its original supervision principle.

SimCLR. We construct two independently augmented views of each observation and optimize the standard NT-Xent loss with temperature 0.1. Augmentations are applied only to the RGB channels, while depth channels remain unchanged. We use color jitter with probability of 0.8, grayscale conversion with probability 0.2, and Gaussian blur with probability of 0.5. For each observation, the two camera views share the same augmentation parameters to preserve cross-view consistency. Two SimCLR views are augmented independently. To preserve the spatial relationships required for manipulation, we do not apply cropping, flipping, rotation, or translation.

SupCon. We convert the continuous task-state descriptor into discrete class labels using K-means clustering. For each task, we divide the samples into two phases and perform clustering separately within each phase. The resulting cluster IDs are used as class labels for SupCon. We use 8 classes for PickCube, OpenDrawer, and PlaceApple, and 3 classes for PushButton. K-means++(Arthur et al., 2007) is performed using at most 100,000 scene-balanced samples per phase, with a maximum of 100 iterations.

RnC. We use the normalized continuous task-state descriptor Z as the regression label. Each descriptor component is normalized according to its task-specific physical scale, and pairwise label distances are computed using the L_1 distance. Following RnC, feature similarity is defined using the negative L_2 distance with temperature $\tau = 2.0$. We use ordinary random minibatches and do not use similarity calibration and cross-environment sampling used by CoCoON. Robot proprioceptive states are not included in the RnC regression labels.

RS-CL. We construct its state-aware supervision from the Franka joint configuration. Specifically, we use the seven arm joint positions and two finger joint positions, normalize them to [-1, 1] using the fixed Panda joint limits, and compute the Euclidean distance between normalized proprioceptive states. The resulting distances are converted into soft target distributions using a temperature β = 1.0. On the visual side, cosine similarity between normalized features is used with temperature τ = 0.2.

R3M. We use the ResNet-18 version of R3M. The inputs are resized to 224×224 using bilinear interpolation and preprocessed with the image normalization scheme corresponding to the R3M model. The two views are independently encoded by the shared R3M encoder, and their output features are concatenated to form the final visual representation for the downstream policy.

HRP. We use HRP as a frozen visual representation encoder. The inputs are resized to 224×224 using bilinear interpolation to match the input resolution required by HRP. We extract CLS feature from each view, concatenate the two features, and project the resulting representation to a 256-dimensional visual feature using a trainable linear adapter. The projected feature is provided to the downstream policy.