# ROBOTS PRE-TRAIN ROBOTS: MANIPULATION-CENTRIC ROBOTIC REPRESENTATION FROM LARGE-SCALE ROBOT DATASET

**Anonymous authors**
Paper under double-blind review

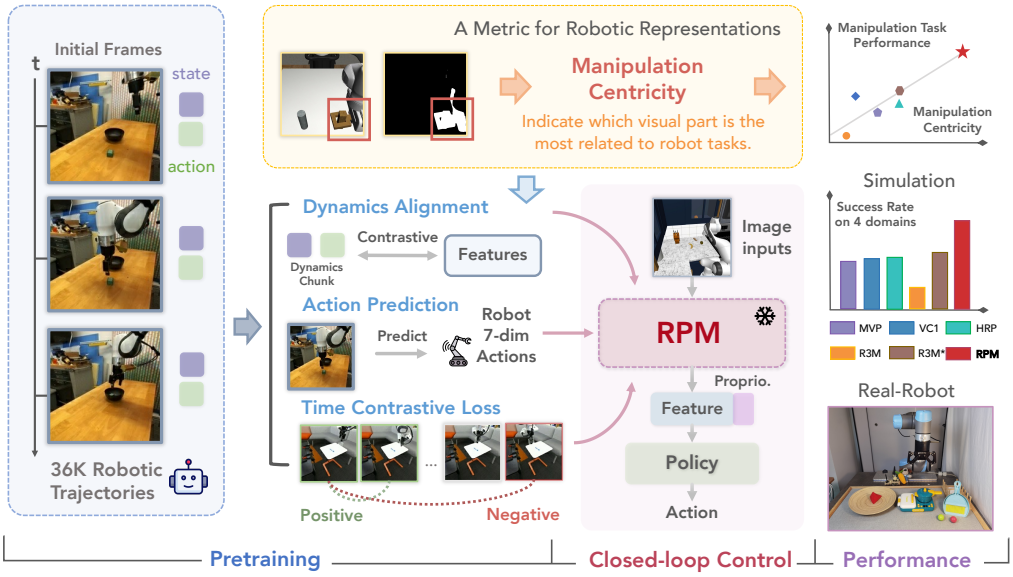



**Figure 1:** **Overview.** We introduce a robotic representation evaluation metric termed *manipulation centricity*, which exhibits a strong correlation with downstream policy performance. Accordingly, we design a new pre-training method, **RPM**, to learn manipulation-centric representation from large-scale robotic datasets. Comprehensive experiments on both simulations and real robot validate the superiority of our proposed representation.

## ABSTRACT

The pre-training of visual representations has enhanced the efficiency of robot learning. Due to the lack of large-scale in-domain robotic datasets, prior works utilize in-the-wild human videos to pre-train robotic visual representation. Despite their promising results, representations from human videos are inevitably subject to distribution shifts and lack the dynamics information crucial for task completion. We first evaluate various pre-trained representations in terms of their correlation to the downstream robotic manipulation tasks (i.e., manipulation centricity). Interestingly, we find that the "manipulation centricity" is a strong indicator of success rates when applied to downstream tasks. Drawing from these findings, we propose **R**obots **P**re-trained with **M**anipulation centricity (**RPM**), a foundation representation learning framework capturing both visual features and the dynamics information such as actions and proprioceptions of manipulation tasks to improve manipulation centricity. Specifically, we pre-train a visual encoder on the DROID (Khazatsky et al., 2024) robotic dataset and leverage motion-relevant data such as robot proprioceptive states and actions. We introduce a novel contrastive loss that aligns visual observations with the robot's proprioceptive state-action dynamics, combined with an action prediction loss and a time contrastive loss during pre-training. Empirical results across four simulation domains with 20 robotic manipulation tasks demonstrate that RPM outperforms the strongest baseline by 14.8%. Additionally, RPM significantly boosts the success rate in three real-world manipulation tasks by 76.9%.

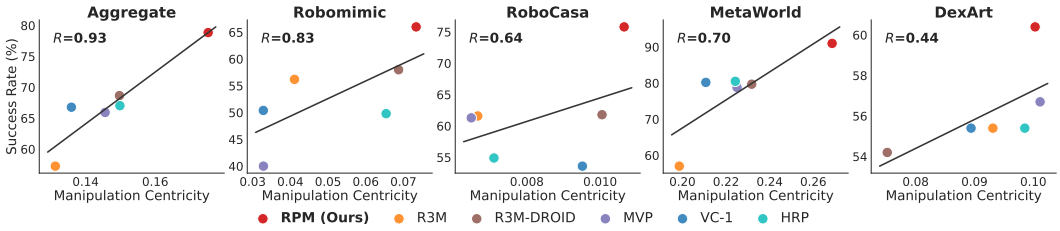



**Figure 2: Correlation between manipulation centricity and downstream performance.** Our findings reveal that (1) the proposed metric of manipulation centricity strongly correlates with the downstream performance of robotic representations, and (2) using the robot dataset DROID (Khazatsky et al., 2024) yields greater benefits for robotic representations than human datasets. (3) These insights motivate our method, RPM, which leverages dynamics labels from the robot dataset to further enhance manipulation centricity and downstream performance. Note that manipulation centricity mainly serves as a valuable comparative metric for ranking different representations within the same domain or aggregated domains.

## 1 Introduction

Grounding robots with generalizable visual representations, termed robotic representation, is crucial for real-world visuomotor control. Pre-training robotic representations on extensive in-domain data offers a promising strategy for robotics, drawing from the success of large-scale pre-training in computer vision (He et al., 2022) and natural language processing (Devlin et al., 2019). However, due to the scarcity of robotics data and high collection costs, recent studies have utilized everyday human operation videos (Grauman et al., 2021; Goyal et al., 2017) to pre-train visual representations for robotic manipulation, positing that knowledge behind human manipulation can inform representations in question. Various levels of human manipulation knowledge — such as task semantics (Nair et al., 2022), pixel-level comprehension (Xiao et al., 2022; Majumdar et al., 2023), and physical interaction (Jia et al., 2024; Srirama et al., 2024)— have demonstrated their effectiveness in benefiting robotic representation. A fundamental question naturally arises: **(Q1) What specific features captured from human data significantly contribute to improving robotic manipulation?**

To investigate this question, we conduct a comprehensive evaluation of prior representations concerning their policy learning performance across various downstream simulation domains. Interestingly, we observe a correlation between a representation's downstream performance and its ability to capture manipulation-relevant regions including robot end-effectors and task-relevant objects. To examine this correlation more formally, we introduce a metric, 'manipulation centricity', and propose an evaluation protocol to quantify it across different representations. The core of this protocol is to measure the similarity between ground truth manipulation regions and the focus of the representation. Our results indicate a strong correlation between downstream performance and manipulation centricity, as illustrated in Figure 2. These findings provide valuable insights into our first question: **(A1) Manipulation centricity emerges as the key factor contributing to enhanced robotic manipulation performance.**

However, human videos not only introduce inherent distribution shifts due to the human-robot embodiment gap, but also lack the dynamics information essential for successful task execution. This naturally leads to the second question: **(Q2) Is there a better dataset choice than human dataset to learn manipulation centricity in robotic representation?** With the recent emergence of large-scale robot datasets (Collaboration et al., 2024; Walke et al., 2023), we hypothesize that the smaller domain gap presented by these datasets may naturally be more suitable for learning manipulation centricity. To prove this hypothesis, we re-train the prior time-contrastive method R3M with a representative robot dataset, DROID (Khazatsky et al., 2024), and indeed observe significant improvements in both performance and manipulation centricity. This answers our second question: **(A2) Large-scale robot datasets can be a better choice than human datasets for learning manipulation centricity.**

Recognizing that robotic datasets provide more relevant information about robot embodiment trajectories, this raises the third question: **(Q3) How to learn manipulation centricity better with large-scale robot datasets?** Our starting point is the dynamics labels, including robot proprioceptive states and actions, in the robot dataset but absent in the human dataset. We consider that these dynamics labels contain the core knowledge behind accomplishing a manipulation task, which has not been explicitly utilized in previous pre-trained robotic representations from human data. To

this end, we introduce a new method, **R**obots **P**re-trained with **M**anipulation centricity (**RPM**), designed to leverage dynamics labels from robot dataset to improve manipulation centricity of robotic representation. Specifically, we propose two objectives: dynamics alignment loss, aligning pixel representations with robot state-action pairs at the same timestep, and action prediction loss, predicting robot actions from image observations. We also incorporate a time contrastive learning objective (Nair et al., 2022) to encode temporal information. Integrating these objectives into the training process significantly improves manipulation centricity, leading to a 15.6% performance increase across four simulation domains encompassing 20 diverse robotic tasks, and a 76.9% improvement across three real-world robot tasks. This answers our third question: **(A3) Effectively utilizing dynamic labels enhances the learning of manipulation centricity.**

We summarize this paper in Figure 1, highlighting three key insights: (1) Manipulation centricity serves as an indicator of the effectiveness of representations for robotic control, as evidenced by the strong correlation with downstream performance. (2) Large-scale robot datasets can be a superior choice compared to human datasets for learning manipulation centricity, as indicated by significant improvements in performance and manipulation centricity when using robot datasets. (3) Effective utilization of dynamics labels from robot datasets significantly enhances the learning of manipulation centricity in robotic representations, demonstrated via the introduction of the RPM method, which incorporates the proposed dynamics alignment and action prediction objectives.

## 2 Experimental Setup: Evaluating Robotic Representations

This section outlines the experimental setup used to evaluate the effectiveness of pre-trained visual representations for robotic manipulation. In line with prior work (Nair et al., 2022; Xiao et al., 2022), we freeze the pre-trained encoders and utilize Imitation Learning (IL; Argall et al. (2009)) for downstream policy learning. Accordingly, the quality of the visual representations is assessed based on IL performance averaged across various downstream tasks.

**Evaluation protocol.** To assess a visual encoder $\mathcal{F}_\phi$, which maps an RGB image $I \in \mathbb{R}^{224\times224\times3}$ to a continuous feature vector $z = \mathcal{F}_\phi(I)$, we introduce a policy network $\pi_\theta$ built atop the frozen encoder $\mathcal{F}_\phi$. This policy network takes as input the feature vector $z$ and the robot's proprioceptive state $s$, outputting an action distribution $\hat{a} \sim \pi_\theta(\cdot|z, s)$. To train the policy, a task-specific demonstration dataset $\mathcal{D}_{IL} = \{\tau_1, \tau_2, ..., \tau_n\}$ is collected, where each demonstration $\tau_i$ is a trajectory consisting of a sequence of expert-level observation-action pairs $\tau_i = \{(I_t, s_t, a_t)\}_{t=1}^{T}$. Then, the policy is optimized via the Behavior Cloning (BC; Bain & Sammut (1995)) algorithm, where the optimization objective is to minimize the error between the predicted action $\hat{a}$ and the optimal action $a$ from demonstration data. For a fair comparison, we employ the same BC algorithm for any used pre-trained encoder in each task. During evaluation, the policy is executed in a closed-loop manner within the environment with online feedback to test its success rate on the target manipulation task. We evaluate at least 20 episodes every fixed number of training epochs, selecting the highest success rate. The mean and standard error of success rates across three seeds are reported.

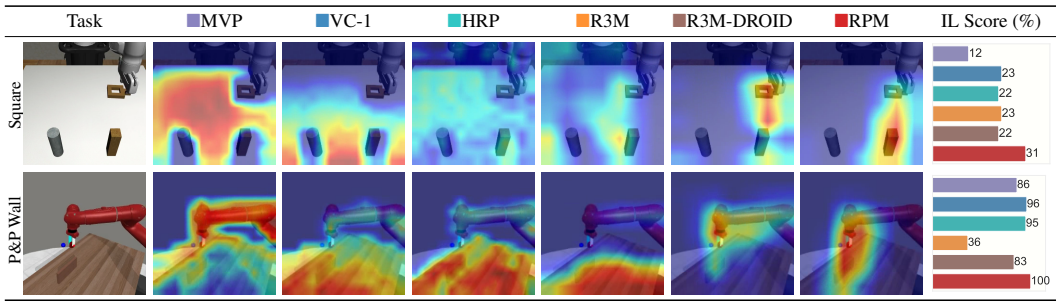
**Simulation environments and datasets.** We select a total of 20 tasks across 4 simulation environments to evaluate representation quality. These tasks encompass a range of end-effectors, including grippers and dexterous hands, along with diverse robot arms and varying levels of manipulation complexity. Task visualizations are shown in Figure 3, with additional details provided in Appendix A.

**Robomimic** **RoboCasa** **MetaWorld** **DexArt**

**Figure 3: Task visualization.** We consider 20 challenging and diverse manipulation tasks spanning 4 domains.

- Robomimic (Mandlekar et al., 2021) features a benchmark suite for tabletop manipulation using the Franka Emika Panda arm and a parallel gripper. We select 3 challenging tasks, utilizing 200 demonstrations from the official proficient human teleoperation dataset for each task.
- RoboCasa (Nasiriany et al., 2024) is a realistic simulation platform focusing on housing scenarios with a Franka Emika Panda arm. We select 3 challenging tasks covering different kitchen scenarios and generate 50 demonstrations via their official scripts for each task.

**Table 1: Grad-CAM visualization.** We present Grad-CAM visualizations alongside their corresponding task success rates for the Square task from Robomimic and the Pick Place Wall task from MetaWorld. Representations that capture the robot's end-effectors and task-relevant objects are linked to improved downstream performance. Numerical results of this table is provided in Appendix A.12. Comprehensive visualizations can be found in Appendix A.2.




- MetaWorld (Yu et al., 2019) features a benchmark suite of robotic tabletop manipulation with the Sawyer robotic arm and gripper. We select 10 diverse tasks according to the difficulty categorized in Seo et al. (2022) and collect 25 demonstrations via official scripted controllers for each task.
- DexArt (Bao et al., 2023) features a benchmark suite for dexterous manipulation tasks on articulated objects using the XArm6 arm and an Allegro hand. We select all 4 tasks and gather 100 demonstrations rollout by well-trained RL policies for each task.

**Comparison methods.** We consider representative methods in robotic visual representation:

- MVP (Xiao et al., 2022) pre-trains a Vision Transformer (ViT, Dosovitskiy et al. (2021)) using Masked Auto-Encoding (MAE, (He et al., 2022)) on a mix of human-object interaction videos.
- VC-1 (Majumdar et al., 2023) is trained similarly to MVP but also additionally incorporates navigation and the ImageNet (Deng et al., 2009) dataset during pre-training.
- HRP (Srirama et al., 2024) fine-tunes a pre-trained ViT to predict human affordance labels, including hand pose, contact points, and active objects extracted from human videos.
- R3M (Nair et al., 2022) pre-trains a ResNet (He et al., 2015) model on human videos using time contrastive learning and video-language alignment to extract temporal and semantic information.

## 3 Investigation: Manipulation Centricity of Representations

Prior work utilizes in-the-wild human videos to assist robotic representation learning, yet the domain gap between human and robotic tasks may influence the representation quality. To investigate this problem, we take an initial step by visualizing these representations in simulated tasks. Interestingly, we observe that a representation's downstream task performance appears to correlate with its ability to capture manipulation-relevant regions. To investigate this correlation more formally, we introduce a metric, *manipulation centricity*, and propose an evaluation protocol to measure it across different methods. Our results indicate a strong correlation between manipulation centricity and downstream task performance, which strongly guides our method design in the next section. We introduce the whole pipeline in the following parts and present more details in Appendix B.

**Feature visualization & motivation.** We use Gradient-weighted Class Activation Mapping (Grad-CAM; Selvaraju et al. (2017)), a widely-adopted network visualization technique in computer vision, to analyze the features of existing representations. Grad-CAM highlights the regions of an input image that are most influential in the model's decision-making process. In our context, this technique reveals how the network interprets images within manipulation tasks, as shown in Table 1. Specifically, in the Square task, we observe that the MVP model tends to focus on irrelevant regions, such as the table, rather than the manipulator or objects, while in the Pick Place Wall task, R3M exhibits a similar pattern. Both cases correspond to poor downstream performance. In contrast, representations that emphasize the robot's end-effectors and objects are associated with better downstream performance. This motivates our investigation into whether representation quality is linked to its ability to capture these 'manipulation-relevant' regions, a property we define as manipulation centricity.

**Measuring Manipulation Centricity.** To quantify this metric, we compute the similarity between the regions highlighted by Grad-CAM and the ground truth regions corresponding to the end-effector and task-relevant objects. The ground truth is generated using the powerful segmentation model SAM 2 (Ravi et al., 2024), with manual annotation of key points within the target regions. This approach enables efficient annotation across an entire video of task execution. We apply this annotation method to create segmentation masks for demonstration data $D_{IL}$ across all simulated tasks, forming our evaluation dataset. As illustrated in Figure 4, manipulation centricity is then measured by averaging the Jaccard Index, a widely-used similarity metric in image segmentation also known as Intersection-Over-Union (IoU), between the binarized Grad-CAM outputs and the ground truth segmentation masks over the entire evaluation dataset, which is computed through the following formula:

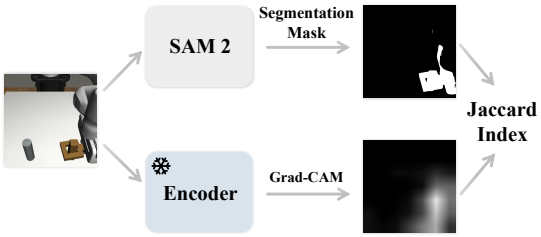



**Figure 4: Measurement of manipulation centricity.**

$$\mathrm{MC}(\mathcal{F}_\phi) = \mathbb{E}_{I\sim D_{IL}}\left[\mathrm{IoU}\left(\mathcal{F}_{\mathrm{SAM2}}(I), \text{Grad-CAM}(\mathcal{F}_\phi(I))\right)\right], \tag{1}$$

where $\mathcal{F}_{\mathrm{SAM2}}(I)$ denotes the segmentation mask predicted by SAM 2.

**Key Findings.** The aggregate results, presented in Figure 2, reveal a strong correlation between manipulation centricity and downstream task performance, with a Pearson correlation coefficient of $R = 0.93$. We also evaluate this correlation within individual simulation domains, where the evaluation dataset consists of demonstrations specific to each domain. The positive correlation remains consistent across all domains, albeit with varying strengths. In summary, these results suggest that manipulation centricity measured on our entire evaluation dataset is a reliable indicator of the effectiveness of robotic representations.

## 4 RPM: Learning Manipulation-Centric Representation

Drawing from the conclusions in Section 3, our focus shifts to improving the manipulation centricity of robotic representations in this section. This is achieved by two attempts in our proposed method, **R**obots **P**re-trained with **M**anipulation centricity (**RPM**). First, we find that re-training existing models with large-scale robot data (*i.e.*, DROID (Khazatsky et al., 2024)) instead of human data can significantly improve manipulation centricity. This indicates the value of utilizing robot-specific datasets, as discussed in Section 4.1. Second, we introduce novel pre-training objectives that leverage the robot's state-action dynamics—information provided in robot data, which is inherently absent in human datasets, to further enhance manipulation centricity, detailed in Section 4.2.

### 4.1 Large-Scale Robot Dataset

**Dataset processing.** In recent years, several large-scale robot datasets have been introduced (Brohan et al., 2023; Walke et al., 2023; Collaboration et al., 2024; Khazatsky et al., 2024). Among these, we select the DROID dataset (Khazatsky et al., 2024) for our method because of its extensive scene diversity and a large volume of data. The dataset is collected using the Franka robot arm and Robotiq 2F-85 gripper via teleoperation, comprising a total of 76k trajectories. Each trajectory includes RGB images from two external Zed 2 cameras, robot proprioceptive states, and actions consisting of delta 6D poses and 1-DoF gripper actions. To ensure data quality, we filter out trajectories with fewer than 40 timesteps to ensure adequate temporal information in each trajectory. Additionally, trajectories containing incomplete or single-word language instructions are removed, as these may indicate lower-quality interactions. After processing, we retain 36k trajectories for pre-training.

**A motivating discovery.** Many previous methods rely on human-object interaction datasets to learn manipulation concepts from human behavior (Nair et al., 2022; Xiao et al., 2022; Srirama et al., 2024). However, the domain gap between human hands and robotic manipulators may inherently affect representation quality. We re-train the R3M model using the robot-specific DROID dataset, yielding a variant we call R3M-DROID (equivalent to R3M*). As expected, Grad-CAM visualizations in Table 1 show that R3M-DROID better captures manipulation-relevant regions, and quantitative results, as shown in Figure 2, confirm this improvement. This suggests that robot-specific data inherently benefits representation learning by narrowing the domain gap between training and deployment environments.

### 4.2 TRAINING RPM

Unlike human datasets, robot datasets provide access to manipulator dynamics, including proprioceptive states and actions. While previous approaches don't utilize this dynamics information, we hypothesize that incorporating it will enhance manipulation centricity. To achieve this, we design two new training objectives that explicitly leverage robot dynamics. Additionally, we adopt a semantic learning loss from prior work (Nair et al., 2022) to retain semantic information in the representations. These objectives and their integration into our training process are detailed below.

**Dynamics alignment.** Our first insight is that each image observation corresponds to an underlying proprioceptive robot state at every timestep. We aim to learn this correspondence, referred — to as state-action dynamics alignment —through contrastive learning. Formally, we define a state-action dynamic chunk of length $l$ at timestep $t$ as $d_t = [s_{\lceil t-\frac{l}{2} \rceil}, a_{\lceil t-\frac{l}{2} \rceil}, s_{\lceil t-\frac{l}{2}+1 \rceil}, \ldots, s_{\lfloor t+\frac{l}{2} \rfloor}]$. The positive sample for $d_t$ is its corresponding RGB image $I_t$ at the same timestep, while the negative samples are drawn from a different timestep $k$ within the same trajectory. During training, we randomly choose $t$ and $k$. The encoder $\mathcal{F}_\phi$ is trained to discard irrelevant details from high-dimensional images and retain the essential information for manipulation. We employ the InfoNCE loss (van den Oord et al., 2019) for contrastive learning, introducing an MLP projector $H$ to map the dynamics chunk $d_t$ to the same dimension as the image feature vector $z = \mathcal{F}_\phi(I)$.

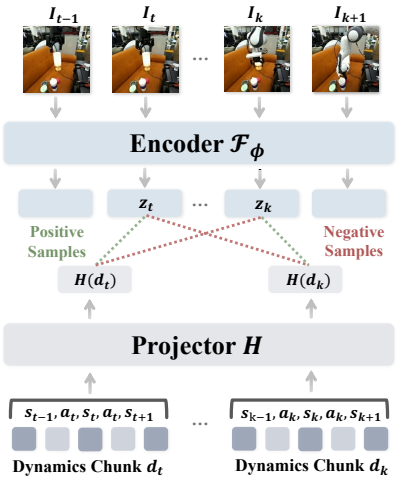



**Figure 5: Illustration of objective $\mathcal{L}_{\text{dyn}}$.**

The objective function is illustrated in Figure 5 and formalized as:

$$\mathcal{L}_{\text{dyn}} = -\sum_{b \in \mathcal{B}} \log \frac{e^{\mathcal{S}(z_t^b, H(d_t)^b)}}{e^{\mathcal{S}(z_t^b, H(d_t)^b)} + e^{\mathcal{S}(z_t^b, H(d_k)^b)}}, \tag{2}$$

where $\mathcal{S}$ represents the negative L2 distance, and $b$ denotes a sample from the batch $\mathcal{B}$.

**Action prediction.** We also integrate a behavior cloning (BC)-like actor into our pre-training framework, based on the idea that robotic representations should be predictive of expert-level behaviors in the dataset. The actor is implemented as a shallow MLP head that maps the image feature vector $z_t$ to the predicted robot actions $\hat{a}t$. We use mean squared error as the objective for action prediction:

$$\mathcal{L}_{\text{act}} = -\sum_{b \in \mathcal{B}} \text{MSE}(a_t^b, \hat{a}_t^b). \tag{3}$$

**Temporal contrast.** We also wish the representation to encode temporal information, which has shown importance for manipulation tasks (Zhao et al., 2023). To this end, we adopt the time-contrastive learning objective from Nair et al. (2022), which encourages temporally close frames in a video to be closer in the embedding space than those that are temporally distant or from different videos. For this, we sample a frame triplet $(I_u, I_v, I_w)$ where $u < v < w$, and compute the following loss:

$$\mathcal{L}_{\text{tcl}} = -\sum_{b \in \mathcal{B}} \log \frac{e^{\mathcal{S}(z_u^b, z_v^b)}}{e^{\mathcal{S}(z_u^b, z_v^b)} + e^{\mathcal{S}(z_u^b, z_w^b)} + e^{\mathcal{S}(z_u^b, z_u^{\neq b})}} \tag{4}$$

where $z_u^{\neq b}$ is a negative sample from a different video within the batch $\mathcal{B}$.

**Oveall objective & implementations.** We train RPM with a combination of introduced objectives:

$$\mathcal{L}_{\text{RPM}} = \mathcal{L}_{\text{dyn}} + \mathcal{L}_{\text{act}} + \mathcal{L}_{\text{tcl}}. \tag{5}$$

We do not introduce additional hyperparameters for weighting these objectives, as $\mathcal{L}_{\text{PRM}}$ already yields strong empirical performance. Our default encoder backbone is ResNet-50, and we use Adam (Kingma, 2015) as the optimizer. The encoder is trained for 500k steps to ensure convergence, and we select the last checkpoint as the final model. The whole training process is used **50 hours with a single Nvidia 3090**. Further training details are provided in Appendix A.

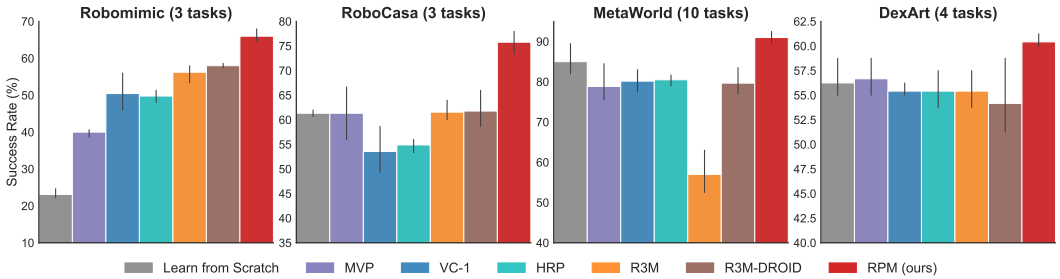



**Figure 6: Simulation results.** We evaluate RPM and baselines across different domains. Our method consistently outperforms the baselines. Results are mean success rate aggregated over 3 seeds with standard deviation.

## 5 Evaluation and Analysis of RPM

Our experiments are conducted in simulation and the real world to answer the following questions:

**(1)** Does RPM learn manipulation centricity and outperform baselines in simulation? (Section 5.1)

**(2)** Can the conclusions from **(1)** generalize to real-world manipulation tasks? (Section 5.2)

**(3)** Which design choices of RPM matter for learning manipulation centricity? (Section 5.3)

**(4)** What benefits does large-scale robot data bring to robotic representation learning? (Section 5.4)

### 5.1 Simulation Results

**RPM outperforms baselines on simulation tasks.** Thanks to its improved manipulation centricity, RPM delivers substantial downstream performance gains compared to the strongest baseline methods across all selected domains. This holds even in the DexArt domain, which involves a dexterous hand as the end-effector—a setup different from the gripper used in the DROID pre-training dataset. Moreover, in the MetaWorld domain, where prior baselines struggle and perform no better than the Learning-from-Scratch (LfS) method, our method maintains a significant advantage. We attribute the poor performance of baselines in MetaWorld to the limited number of demonstrations, which makes it difficult for policy to leverage the pre-trained representations. In contrast, RPM reduces the policy's learning burden by providing manipulation-centric features that better fit the task.

**RPM does improve manipulation centricity.** We begin by presenting visualizations of Grad-CAM results for RPM in Table 1 (see all results in Table 14). Qualitatively, our representation excels at capturing manipulation-relevant regions of each task. For instance, in the Robomimic Square task, RPM focuses on the gripper, square tool, and the target area—key elements of the task. In contrast, other methods like VC-1 and MVP either fail to highlight these areas, or, like R3M and HRP, only capture them partially. This improved localization of task-relevant features is further confirmed quantitatively in Figure 2, where RPM significantly enhances manipulation centricity across all domains. Notably, RPM even highlights the optimal path the end-effector should follow, suggesting that our representation also learns essential information related to task execution.

### 5.2 Real Robot Results

**Experimental setup.** As visualized in Figure 7, our real-world experiments involve a UR5e arm equipped with a Robotiq 2F-85 gripper and a RealSense D435i camera for RGB image capture. We designed three tabletop manipulation tasks with different objects and manipulation skills:

- Lift. The robot grips the sandbag on the plate and lifts it up in the sky.
- Sweep. The robot grasps the broom to sweep the trash from the table into the dustpan.
- Rearrange. The robot picks up the on-table pot and places it at the designated spot on the stove.

The demonstrations used for BC-based policy training are collected by keyboards with the human operator, with 30 collected for Lift and 40 for more difficult tasks Rearrange and Sweep. The pre-trained representations are frozen during policy training inheriting the simulation setup. For a fair comparison, we evaluate each method with the same sets of start-up conditions, which are unseen in demonstrations, in each task. More experimental details can be found in Appendix A.

**RPM significantly surpasses baselines on real robot tasks.** The evaluation results, shown in Table 2, demonstrate that RPM consistently outperforms the baseline methods across all tasks. In

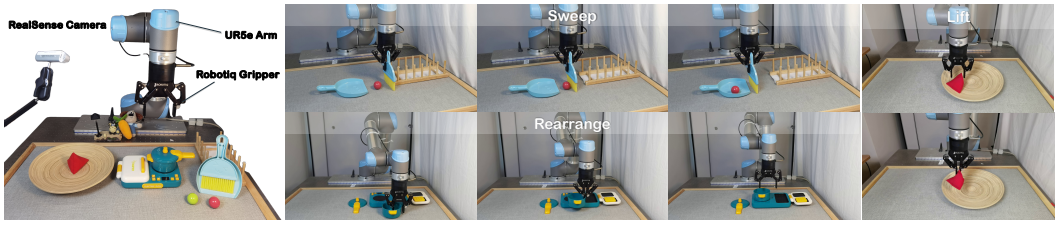



**Figure 7: Real robot setup.** We design 3 real-world robot tasks with different manipulation skills and objects.

the Lift and Rearrange tasks, baseline methods often fail to grasp object accurately, particularly when object positions were unseen in the demonstrations. In the Sweep task, their performance deteriorates further due to poor accuracy in sweeping, with the trash frequently missing the dustpan and rolling off in unintended directions. In contrast, the policies trained with our representation exhibit robust and generalizable handling of these complex tasks.

| Task | LfS | MVP | VC-1 | R3M | RPM |
|---|---|---|---|---|---|
| Lift | 5/10 | 6/10 | 5/10 | 6/10 | **9/10** |
| Sweep | 3/10 | 1/10 | 2/10 | 1/10 | **7/10** |
| Rearrange | 2/10 | 3/10 | 6/10 | 4/10 | **7/10** |
| All | 10/30 | 10/30 | 13/30 | 11/30 | **23/30** |

**Table 2: Real robot results.** Our method RPM performs best in all tasks. Each method is fairly assessed over 10 trials on each task.

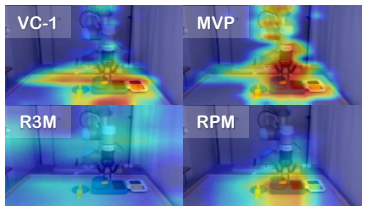



**Figure 8: Grad-CAM on Rearrange.** RPM is with best manipulation centricty.

We also visualize the Grad-CAM results for each method on the challenging Rearrange task in Figure 8. Representations such as R3M fail to focus on key manipulation regions, resulting in poor performance. MVP and VC-1 perform better, as they can capture the robot arm and relevant objects on the table, but they tend to over-focus on irrelevant items like the pot. Our method avoids these pitfalls, achieving the best performance across all tasks. In conclusion, these real-world experiments underscore the effectiveness of our method in enhancing robot manipulation capabilities, significantly outperforming baseline approaches in challenging scenarios.

### 5.3 Ablation Studies

The results presented in Table 4 indicate the effect of key design choices in RPM, offering insights and guidance when employing our manipulation-centric representations. All experiments are conducted on three challenging tasks: Can from Robomimic, Stick Pull from MetaWorld, and Laptop from DexArt, covering all robot arms and end-effectors in prior simulation experiments.

**All objectives improve manipulation centricity.** We begin by evaluating the effectimpact of each training objective outlined in Equation (5) through an ablation study. Our results reveal that all objectives are essential to achieving strong downstream performance. In particular, the dynamics alignment and action prediction losses have the most significant effect. We attribute this to their ability to effectively utilize dynamics-relevant information in the robot dataset, which enhances manipulation centricity. Additionally, the temporal contrastive loss also plays a crucial role by capturing the temporal dependencies in the video sequences. Collectively, this analysis supports our key claim: learning manipulation-centric representations is beneficial for robotic manipulation, and our proposed objectives substantially improve this capability.

**Table 4: Key design choices of RPM.**

| Ablated Components | Success Rate (%) |
|---|---|
| **Training Objective** | |
| w/o. objective $\mathcal{L}_{\text{dyn}}$ | 66.2 ($\pm$0.8) |
| w/o. objective $\mathcal{L}_{\text{act}}$ | 71.3 ($\pm$1.2) |
| w/o. objective $\mathcal{L}_{\text{tcl}}$ | 72.0 ($\pm$1.2) |
| **Dynamic Chunk** | |
| Length $l$: 3$\rightarrow$1 | 72.1 ($\pm$2.9) |
| Length $l$: 3$\rightarrow$5 | 76.8 ($\pm$2.4) |
| Length $l$: 3$\rightarrow$7 | 76.8 ($\pm$2.2) |
| **Encoder Backbone** | |
| ResNet-: 50$\rightarrow$18 | 77.3 ($\pm$1.8) |
| ResNet-: 50$\rightarrow$34 | 77.9 ($\pm$1.7) |
| RPM (original) | **83.2** ($\pm$1.3) |

**Medium dynamic chunk length works best.** Recall that dynamic chunks define the temporal horizon of robot state-action pairs utilized for modeling robot dynamics, as specified in Equation (2). Our experiments indicate that a medium chunk length of 3 yields optimal performance. In contrast, a shorter chunk length (i.e., $l = 1$) fails to adequately capture action information, resulting in an insufficient representation of the underlying dynamics. Additionally, the size of a single-state chunk is considerably smaller than that of the pre-trained visual features, which can introduce noise when projected through a multi-layer perceptron (MLP). On the other hand, excessively long chunks may

complicate the accurate modeling of dynamics. This study highlights the necessity of effectively encoding dynamic information in representations to enhance manipulation centricity.

**Larger encoders lead to better performance.** Finally, we also observe a clear trend: larger encoders consistently result in better performance. We hypothesize that larger models have a greater capacity to comprehend complex scenes and capture critical information related to dynamics and manipulation centricity. This finding is consistent with the ones in previous work (Shang et al., 2024), suggesting that our method scales well with the model size and has the potential for even greater improvements in performance if larger models are employed.

### 5.4 Analysis on Robot Dataset

Besides methodology design, we also reveal insights into the robot dataset in robotic representations.

**Larger dataset, better performance.** We begin by examining how the size of the robot dataset affects pre-training outcomes. Specifically, we reduced the data in each DROID scene from 100% to 25% and assessed the downstream performance, as shown in Figure 9. Our findings indicate that larger datasets contribute to improved representation quality. This seemingly contrasts with previous research (Dasari et al., 2023), which suggested that merely increasing dataset size may not yield benefits. We remark that the key differences here are (1) our use of robot data rather than human data, and (2) the incorporation of additional robot dynamics information. These factors enhance the scalability and effectiveness of RPM when applied to larger robot datasets. While some studies have attempted to combine human and robot data (Dasari et al., 2023; Majumdar et al., 2023), little improvement was observed, likely due to the insufficient utilization of robot dynamics. In summary, our method effectively scales with dataset size by leveraging dynamics information.

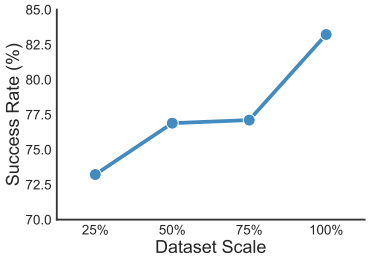



**Figure 9: Effect of robot dataset size.**

**Greater benefits for tasks with less embodiment gap.** Next, we investigate which downstream tasks benefit more from pre-training with robot data. Specifically, we examine the embodiment gap associated with the end-effector. The simulation tasks are accordingly categorized into gripper-based and dexterous hand-based tasks. As illustrated in Figure 10, both R3M-DROID and R3M outperform R3M on gripper-based tasks in terms of manipulation centricity and downstream success rate, supporting our earlier conclusions. However, performance drops in hand-based tasks, where R3M-DROID even underperforms R3M. This is likely due to the fact that all data in DROID was collected using a gripper, which presents an embodiment gap compared to dexterous hands. To alleviate this issue, we suggest two attempts in the future: (1) leveraging dynamics from robot data better to mitigate the embodiment gap, and (2) incorporating more end-effectors, such as dexterous hands, into robot datasets to enhance our manipulation-centric representations.

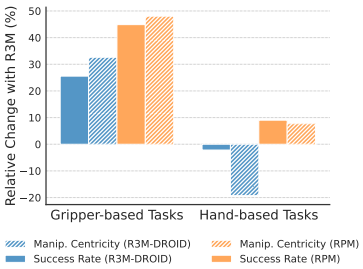



**Figure 10: Downstream domain gap.**

**Feature analysis.** To gain a more intuitive understanding of the impact of the robot dataset, we employ t-SNE (Van der Maaten & Hinton, 2008) to process and visualize feature embeddings generated by the pre-trained encoder. The visualization results are presented in Figure 11. In the simulation domain, R3M struggles to cluster images within individual tasks. However, this issue is partially alleviated by using a robot dataset, as R3M-DROID exhibits improved clustering ability. Nonetheless, due to the significant domain gap between DROID and simulation environments, it still encounters difficulties in distinguishing many tasks. In contrast, our method demonstrates markedly superior clustering capabilities, indicating the importance of incorporating real-robot dynamics for effective robotic representation. Additionally, we observe that all three methods exhibit good clustering performance in real-world tasks, likely attributed to the more distinct visual changes between tasks compared to simulations. However, R3M remains inferior to the other two methods, reinforcing the critical role of robot datasets in enhancing robotic representations.

## 6 Related Work

**Pretrained robotic representations.** The development of pre-trained visual representations (PVRs) has improved the efficiency of downstream policy learning in robotics. Notable approaches include

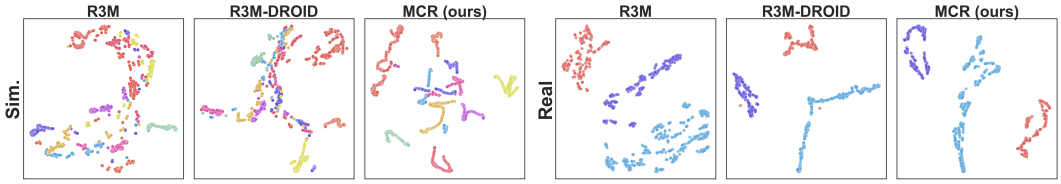



**Figure 11: t-SNE visualization.** We do t-SNE visualization on 10 simulation tasks from MetaWorld and 3 real robot tasks. Each dot represents an image frame and each color indicates a task. The results demonstrate that (1) our representation has the best clustering ability and (2) robot data is helpful to robotic representation.

a variant of MoCo-v2 (Parisi et al., 2022) that integrates multi-layer features, MVP (Xiao et al., 2022) and VC-1 (Majumdar et al., 2023) utilizing Masked Autoencoders (He et al., 2022), and R3M (Nair et al., 2022), which employs a time-contrastive objective and video-language alignment. Other works like Zheng et al. (2024) use temporal action contrastive learning, while MPI (Jia et al., 2024) focuses on predicting transition frames based on language instructions. HRP (Srirama et al., 2024) extracts affordances from large-scale human videos for enhanced generalization, and Theia (Shang et al., 2024) distills diverse vision models for robot learning. VIP (Ma et al., 2023b) generates dense reward functions for robotic tasks. LIV (Ma et al., 2023a) incorporates language to learn a multi-modal representation and reward. POCR (Shi et al., 2024) pre-trains an object-centric representation and HODOR (Athar et al., 2022) learns high-level object descriptors. Similar to our method, RPT (Radosavovic et al., 2023) employs dynamics labels for representation. In contrast, our work introduces the concept of manipulation centricity, leveraging large-scale robotic data to capture manipulation-specific dynamics, resulting in improved performance on downstream tasks.

**Learning from large-scale robotic data.** Recent advancements in robotics increasingly utilize large-scale datasets to enhance the capabilities of robotic systems. Collaboration et al. (2024) introduces the Open X-Embodiment dataset, the largest robotic dataset comprising extensive data from diverse robot embodiments across various tasks. The RT-X model (Collaboration et al., 2024), trained on this diverse dataset, shows promising results in cross-robot skill transfer as a generalist. Team et al. (2024) introduces Octo, a transformer-based diffusion policy also trained on Open X-Embodiment, supporting flexible task and observation definitions. Additionally, OpenVLA (Kim et al., 2024) is developed as a vision-language model enabling direct mapping from image inputs to continuous robot actions. Unlike these approaches, our work focuses on extracting dynamics and interaction information from robotic datasets to create specialized visual representations for policy learning, offering an efficient alternative to generalist policies.

**Factors driving effectiveness in robotic representation.** Burns et al. (2024) identifies emergent segmentation ability as a critical factor in the success of pre-trained visual models for robotic manipulation. While their focus is on generalization capability, our emphasis is on the downstream learning efficiency of robotic representations, investigating this from the perspective of manipulation centricity. Additionally, Theia (Shang et al., 2024) highlights the importance of high entropy in feature norm distribution for performance enhancement. Although they validate their findings in simulation locomotion tasks, we evaluate the effectiveness of manipulation centricity in both simulated and real-world robot manipulation tasks.

## 7 Conclusions

Our work introduces the concept of manipulation centricity in visual representations for robotic manipulation, revealing its crucial role in downstream task performance. By leveraging large-scale robot data and dynamics-aware learning objectives, we develop a method that significantly enhances the extraction of manipulation-centric features. This approach not only advances the state-of-the-art in robotic manipulation across diverse tasks but also provides a new lens through which to understand and evaluate representation learning in robotics. Our findings highlight the importance of aligning representation learning with the specific demands of robotic control, potentially shifting the paradigm of how we approach feature extraction for embodied agents. Looking forward, this work opens avenues for exploring multi-modal integration, such as using language instructions to learn task-aware features and further leveraging trajectory data to capture spatial-temporal robotic dynamics, promising to further narrow the gap between pre-trained generalized representation models and real-world robotic manipulation.

## LIMITATION AND FUTURE DIRECTIONS

While our method demonstrates strong performance, we acknowledge several key limitations. Our current approach faces challenges with cluttered environments where multiple objects may affect manipulation centricity clarity. The computation of manipulation centricity requires human annotation effort, and our framework may have limitations in handling long-horizon complex control tasks. Future work includes exploring language instructions for automated manipulation centricity computation and extending training to the larger robotic datasets, such as Open-X-Embodiment, for better generalization across diverse robot embodiments. Furthermore, applying and refining our method for high-dynamics robotics tasks, such as object throwing, would provide valuable insights and facilitate evaluation on more real-world robotic takss. Annotating the manipulation-centric components of large-scale robotic datasets and leveraging this data to directly enhance manipulation-centricity represent another promising avenue. These directions aim to further strengthen the practicality, scalability, and generalizability of our approach, advancing its applicability to a broader range of robotic tasks and environments.

## REFERENCES

Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. Robotics and Autonomous Systems, 2009.

Ali Athar, Jonathon Luiten, Alexander Hermans, Deva Ramanan, and Bastian Leibe. Hodor: High-level object descriptors for object re-segmentation in video learned from static images. In Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

Michael Bain and Claude Sammut. A framework for behavioural cloning. In Machine Intelligence 15, 1995.

Chen Bao, Helin Xu, Yuzhe Qin, and Xiaolong Wang. Dexart: Benchmarking generalizable dexterous manipulation with articulated objects. In Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. Robotics: Science and Systems (RSS), 2023.

Kaylee Burns, Zach Witzel, Jubayer Ibn Hamid, Tianhe Yu, Chelsea Finn, and Karol Hausman. What makes pre-trained visual representations successful for robust manipulation? In Conference on Robot Learning (CoRL), 2024.

Hongyi Chen, Abulikemu Abuduweili, Aviral Agrawal, Yunhai Han, Harish Ravichandar, Changliu Liu, and Jeffrey Ichnowski. Korol: Learning visualizable object feature with koopman operator rollout for manipulation. In Conference on Robot Learning (CoRL), 2024.

Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun,

Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. In International Conference on Robotics and Automation (ICRA), 2024.

Sudeep Dasari, Mohan Kumar Srirama, Unnat Jain, and Abhinav Gupta. An unbiased look at datasets for visuo-motor pre-training. In Conference on Robot Learning (CoRL), 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In North American Chapter of the Association for Computational Linguistics (NAACL), 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In International Conference on Learning Representations (ICLR), 2021.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In International Conference on Computer Vision (ICCV), 2017.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh K. Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Z. Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Kahsay Gebreselasie, Cristina González, James M. Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolár, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu,

Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran K. Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbeláez, David J. Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard A. Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video. In Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In International Conference on Machine Learning (ICML), 2020.

Tianying Ji, Yongyuan Liang, Yan Zeng, Yu Luo, Guowei Xu, Jiawei Guo, Ruijie Zheng, Furong Huang, Fuchun Sun, and Huazhe Xu. Ace: Off-policy actor-critic with causality-aware entropy regularization. arXiv preprint arXiv:2402.14528, 2024.

Zeng Jia, Bu Qingwen, Wang Bangjun, Xia Wenke, Chen Li, Dong Hao, Song Haoming, Wang Dong, Hu Di, Luo Ping, Cui Heming, Zhao Bin, Li Xuelong, Qiao Yu, and Li Hongyang. Learning manipulation by predicting interaction. Robotics: Science and Systems (RSS), 2024.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset. Robotics: Science and Systems, 2024.

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.

Diederik P Kingma. Adam: A method for stochastic optimization. In International Conference on Learning Representations (ICLR), 2015.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. CURL: Contrastive unsupervised representations for reinforcement learning. In International Conference on Machine Learning (ICML), 2020.

Yongyuan Liang, Tingqiang Xu, Kaizhe Hu, Guangqi Jiang, Furong Huang, and Huazhe Xu. Make-an-agent: A generalizable policy network generator with behavior-prompted diffusion. arXiv preprint arXiv:2407.10973, 2024.

Cong Lu, Philip J Ball, Tim GJ Rudner, Jack Parker-Holder, Michael A Osborne, and Yee Whye Teh. Challenges and opportunities in offline reinforcement learning from visual observations. Transactions on Machine Learning Research (TMLR), 2023.

Yecheng Jason Ma, William Liang, Vaidehi Som, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. In International Conference on Machine Learning (ICML), 2023a.

Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In International Conference on Learning Representations (ICLR), 2023b.

Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Pieter Abbeel, Jitendra Malik, Dhruv Batra, Yixin Lin, Oleksandr Maksymets, Aravind Rajeswaran, and Franziska Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? In International Conference on Neural Information Processing Systems (NeurIPS), 2023.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In Conference on Robot Learning (CoRL), 2021.

Bogdan Mazoure, Remi Tachet des Combes, Thang Long Doan, Philip Bachman, and R Devon Hjelm. Deep reinforcement and infomax learning. In International Conference on Neural Information Processing Systems (NeurIPS), 2020.

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), 2022.

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.

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 (ICML), 2022.

Ilija Radosavovic, Baifeng Shi, Letian Fu, Ken Goldberg, Trevor Darrell, and Jitendra Malik. Robot learning with sensorimotor pre-training. In Conference on Robot Learning (CoRL), 2023.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714, 2024.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In International Conference on Computer Vision (ICCV), 2017.

Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and P. Abbeel. Masked world models for visual control. In Conference on Robot Learning (CoRL), 2022.

Jinghuan Shang, Karl Schmeckpeper, Brandon B. May, Maria Vittoria Minniti, Tarik Kelestemur, David Watkins, and Laura Herlant. Theia: Distilling diverse vision foundation models for robot learning. In Conference on Robot Learning (CoRL), 2024.

Junyao Shi, Jianing Qian, Yecheng Jason Ma, and Dinesh Jayaraman. Composing pre-trained object-centric representations for robotics from" what" and" where" foundation models. In International Conference on Robotics and Automation (ICRA), 2024.

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

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In International Conference on Machine Learning (ICML), 2021.

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. 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, 2019.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of Machine Learning Research (JMLR), 2008.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In Conference on Robot Learning (CoRL), 2023.

Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. In Conference on Robot Learning (CoRL), 2022.

Guowei Xu, Ruijie Zheng, Yongyuan Liang, Xiyao Wang, Zhecheng Yuan, Tianying Ji, Yu Luo, Xiaoyu Liu, Jiaxin Yuan, Pu Hua, et al. Drm: Mastering visual reinforcement learning through dormant ratio minimization. In International Conference on Learning Representations (ICLR), 2023.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In Conference on Robot Learning (CoRL), 2019.

Zhecheng Yuan, Zhengrong Xue, Bo Yuan, Xueqian Wang, Yi Wu, Yang Gao, and Huazhe Xu. Pre-trained image encoder for generalizable visual reinforcement learning. Advances in Neural Information Processing Systems, 35:13022–13037, 2022.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. Robotics: Science and Systems (RSS), 2024.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. Robotics: Science and Systems (RSS), 2023.

Ruijie Zheng, Xiyao Wang, Yanchao Sun, Shuang Ma, Jieyu Zhao, Huazhe Xu, Hal Daumé III, and Furong Huang. TACO: Temporal latent action-driven contrastive loss for visual reinforcement learning. In International Conference on Neural Information Processing Systems (NeurIPS), 2023.

Ruijie Zheng, Yongyuan Liang, Xiyao Wang, Shuang Ma, Hal Daumé III, Huazhe Xu, John Langford, Praveen Palanisamy, Kalyan Shankar Basu, and Furong Huang. Premier-taco: Pretraining multitask representation via temporal action-driven contrastive loss. In International Conference on Machine Learning (ICML), 2024.

# A More Experimental Details

## A.1 Task Descriptions

We selected a diverse set of tasks from various robotic manipulation benchmarks for evaluation. Specifically, we included three tasks from Robomimic (Mandlekar et al., 2021), three tasks from RoboCasa (Nasiriany et al., 2024), ten tasks from MetaWorld (Yu et al., 2019), and four dexterous tasks from DexArt (Bao et al., 2023). Detailed descriptions of each task are provided below:

- `Can` (Robomimic, $\mathcal{A} \in \mathbb{R}^7$): the task is to manipulate the can using the robot's arm to perform various actions such as picking it up, moving it to a different location, and orienting it in a specific way.
- `Lift` (Robomimic, $\mathcal{A} \in \mathbb{R}^7$): the task is to grasp a specified item and then raise it to a desired height.
- `Square` (Robomimic, $\mathcal{A} \in \mathbb{R}^7$): the task is to pick up a square-shaped nut and place it onto a rod successfully.
- `Close Drawer` (RoboCasa, $\mathcal{A} \in \mathbb{R}^7$): the task is to accurately close a drawer.
- `Coffee Button Press` (RoboCasa, $\mathcal{A} \in \mathbb{R}^7$): the task is to press the button on the coffee machine to start the coffee brewing process.
- `Open Single Door` (RoboCasa, $\mathcal{A} \in \mathbb{R}^7$): the task is to open a door that is singularly paneled, such as a cabinet or microwave door, which is already closed.
- `Assembly` (MetaWorld, $\mathcal{A} \in \mathbb{R}^4$): the task is to grasp a nut and position it on a dowel using the gripper.
- `Bin Picking` (MetaWorld, $\mathcal{A} \in \mathbb{R}^4$): the task is to transfer a disc from one bin container to another.
- `Button Press` (MetaWorld, $\mathcal{A} \in \mathbb{R}^4$): the task is to press a button using a robotic arm to activate a device.
- `Disassemble` (MetaWorld, $\mathcal{A} \in \mathbb{R}^4$): the task is to remove a nut from a peg by picking it up.
- `Drawer Open` (MetaWorld, $\mathcal{A} \in \mathbb{R}^4$): the task is to accurately open a drawer.
- `Hammer` (MetaWorld, $\mathcal{A} \in \mathbb{R}^4$): the task is to drive a screw into the wall using a hammer.
- `Pick Place Wall` (MetaWorld, $\mathcal{A} \in \mathbb{R}^4$): the task is to grab a puck, go around a wall and put the puck in the designated spot.
- `Shelf Place` (MetaWorld, $\mathcal{A} \in \mathbb{R}^4$): the task is to grab a puck and set it on a shelf..
- `Stick Pull` (MetaWorld, $\mathcal{A} \in \mathbb{R}^4$): the task is to use a stick to pull a box by holding onto the stick.
- `Stick Push` (MetaWorld, $\mathcal{A} \in \mathbb{R}^4$): the task is to hold a stick to push a box with it.
- `Bucket` (DexArt, $\mathcal{A} \in \mathbb{R}^{22}$): the task is to elevate a bucket..
- `Faucet` (DexArt, $\mathcal{A} \in \mathbb{R}^{22}$): the task is to activate a faucet using a rotating joint.
- `Laptop` (DexArt, $\mathcal{A} \in \mathbb{R}^{22}$): the task is to grasp the center of the display and then lift the laptop cover.
- `Toilet` (DexArt, $\mathcal{A} \in \mathbb{R}^{22}$): the task is to initiate the process of lifting a bigger toilet seat.

The real-world evaluation protocol is detailed in Section 5.2. To ensure a fair comparison, we evaluate all methods under identical starting conditions that were not present in the demonstration data. While our training data was collected within a confined region, our evaluation protocol introduces significant variations:

For each task, we maintain specific success criteria:

- `Lift`: The robot arm starts from a fixed initial pose, with a sandbag randomly positioned on a fixed base. Success requires lifting the sandbag at least 10 cm above the base.
- `Sweep`: The dustpan, trash, besom, and rack positions are randomly initialized. Success is achieved when the trash is successfully swept into the dustpan using the besom.
- `Rearrange`: The pot and surrounding objects are randomly positioned. Success requires placing the pot precisely on the designated black circular area of the cooking range.

### A.2 More Grad-CAM visualizations

The Grad-CAM visualizations for each task are presented in Table 14, which provides a comprehensive comparison with other baseline methods. Notably, our approach is shown to effectively facilitate the capture of key manipulation-centric features, thereby enhancing the model's ability to focus on the most relevant aspects of the task.

### A.3 Pre-training Hyperparameters

We show our hyperparameters during the pre-training stage in Table 5. Downstream policy learning settings are introduced in section A.5.

**Table 5:** Hyperparameters for RPM pre-training.

| Hyperparameter | Value |
|---|---|
| Encoder type | ResNet50 |
| Batch size | 32 |
| Learning rate | 1e-4 |
| Training steps | 500,000 |
| Data augmentation | RandomResizedCrop (224,(0.5, 1.0)) |
| Optimizer | Adam |
| DROID views used | two exterior views |
| DROID proprioception used | cartesian and gripper position |

### A.4 Pre-training Implementation

Our codebase is built upon the implementation from the official code (Nair et al., 2022). Similarly, for one sample within the batch, we have 5 frames from one video. The initial and final frames are sampled from the first and last 20% of the video, respectively. We employ the same contrastive learning loss implementation, modifying only the positive and negative sample pairs. For more details, please refer to Nair et al. (2022).

To clarify further, we provide our PyTorch-like code implementation of $L_a$ as below:

```
actor_trunk =
   # self.outdim is the output dimension of ResNet, for example, ResNet50
      ↪ is 2048
   nn.Sequential(nn.Linear(self.outdim, 50),
   nn.LayerNorm(50), nn.Tanh())

actor_policy = nn.Sequential(
   # action_dim in our case in 7 for DROID dataset
   nn.Linear(50, 512),
   nn.ReLU(inplace=True),
   nn.Linear(512, 512),
   nn.ReLU(inplace=True),
   nn.Linear(512, action_dim))
```

The MLP part of dynamics alignment loss $L_d$ is implemented as follows:

```
# calculate the length of state-action dynamics chunk
state_input_dim = 14 * self.state_chunk_length # state
state_input_dim += 7 * (self.state_chunk_length - 1) # action

state_encoder = nn.Sequential(
   nn.Linear(state_input_dim, 1024),
   nn.ReLU(),
   nn.Linear(1024, self.outdim))
```

### A.5 Behavior Cloning (BC) Implementation and Settings

**MetaWorld and DexArt.** We evaluate our model on these two domains following (Ze et al., 2024). Scripted policies are used to generate demonstrations in MetaWorld, and policies trained with Rein-

forcement Learning is used in DexArt. We generate 25 demonstrations per task in MetaWorld and 100 in DexArt, where robot end effectors and objects are initially randomized. The downstream BC policy is a three-layer MLP with ReLu activations and hidden sizes of 256. A BatchNorm layer is added before the feature is input into the MLP. The policy is trained with a mean squared error (MSE) loss, a learning rate of 0.001, and a batch size of 256.

**Robomimic.** We adopt the released behavior cloning implementation from RoboMimic, and use their standard imitation learning dataset. The dataset contains 200 demonstrations for each task. We only modify the code to evaluate diverse visual encoders. The downstream policy is a two-layer MLP with hidden sizes of 1024. The policy is trained with an MSE loss, an initial learning rate of 0.001, and a batch size of 100. The learning rate is decayed with a factor of 1.

**RoboCasa.** We utilize the official policy learning implementation from RoboCasa. We use the BC-Transformer algorithm implemented by RoboMimic, with a RoboCasa-standard configuration as reported in the appendix of Nasiriany et al. (2024). We modify the visual observation encoder and train BC policies using the "Human-50" dataset collected by human operators. The policy is trained with an MSE loss, an initial learning rate of 0.0001, and a batch size of 16. The learning rate is decayed with factor 1.

**Real world** Our real-world codebase is adapted from V-D4RL (Lu et al., 2023). We collect demonstrations using a keyboard interface, with 40 demonstrations for Sweep and Rearrange, and 30 for Lift due to its simplicity. The policy is a 3-layer BatchNormMLP with a hidden size of 1024. We train the policy using the log-likelihood loss, with a learning rate of 0.0001 and a batch size of 256.

### A.6 Behavior Cloning (BC) Performance of Each Simulation Task

We show the performance per task of our main results in Table 6. Our method achieves the highest success rate in 19 out of 20 tasks evaluated, showcasing its robustness and adaptability to various scenarios.

**Table 6: Main results on 20 simulation tasks.** Results for each task are provided in this table. A summary across domains is shown in Figure 6.

| | DexArt | | | | Robomimic | | | RobocCasa | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Alg \ Task | Bucket | Faucet | Laptop | Toilet | Can | Lift | Square | CloseDrawer | CoffeeButtonPress | OpenSingleDoor |
| RPM (ours) | **36.7** (±2.9) | **38.3** (±2.9) | **93.3** (±2.9) | 73.3 (±2.9) | **68.0** (±4.0) | **96.0** (±2.3) | **30.0** (±1.2) | **99.3** (±1.2) | **72.0** (±3.5) | **56.0** (±3.5) |
| LfS | 33.3 (±5.8) | 36.7 (±5.8) | 83.3 (±10.4) | 71.7 (±2.9) | 6.0 (±0.0) | 64.0 (±4.2) | 4.0 (±0.0) | 85.3 (±1.2) | 52.0 (±4.0) | 46.7 (±1.2) |
| MVP | 31.7 (±2.9) | 33.3 (±2.9) | 81.7 (±5.8) | **80.0** (±0.0) | 28.0 (±2.0) | 74.0 (±6.4) | 14.0 (±2.3) | 98.0 (±2.0) | 52.7 (±18.9) | 33.3 (±14.5) |
| VC1 | 30.0 (±0.0) | 35.0 (±0.0) | 85.0 (±0.0) | 71.7 (±2.9) | 44.0 (±7.0) | 74.0 (±9.2) | 20.0 (±3.5) | 98.7 (±2.0) | 29.3 (±5.8) | 33.3 (±7.0) |
| R3M | 31.7 (±2.9) | 36.7 (±2.9) | 81.7 (±5.8) | 71.7 (±2.9) | 50.0 (±4.2) | 86.0 (±6.0) | 24.0 (±1.2) | 88.7 (±3.1) | 47.3 (±6.1) | 48.7 (±7.6) |
| HRP | 31.7 (±2.9) | 36.7 (±2.9) | 90.0 (±5.0) | 63.3 (±14.4) | 42.0 (±3.5) | 86.0 (±3.5) | 26.0 (±2.3) | 91.3 (±4.6) | 35.3 (±11.6) | 38.0 (±6.0) |
| R3M-Droid | 35.0 (±5.0) | 33.3 (±2.9) | 80.0 (±0.0) | 66.7 (±7.6) | 54.0 (±2.3) | **96.0** (±0.0) | 22.0 (±3.1) | 88.7 (±2.3) | 51.3 (±2.3) | 45.3 (±7.6) |

| | MetaWorld | | MetaWorld (Medium) | | MetaWorld (Hard) | MetaWorld (Very Hard) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Alg \ Task | Button Press | Drawer Open | Bin Picking | Hammer | Assembly | Shelf Place | Disassemble | Stick Pull | Stick Push | Pick Place Wall |
| RPM (ours) | **100.0** (±0.0) | **100.0** (±0.0) | **96.7** (±2.9) | **100.0** (±0.0) | **100.0** (±0.0) | **41.7** (±5.8) | **93.3** (±2.9) | **86.7** (±2.9) | **100.0** (±0.0) | **91.7** (±2.9) |
| LfS | 96.7 (±2.9) | 95.0 (±5.0) | 81.7 (±2.9) | 95.0 (±5.0) | 95.0 (±5.0) | 35.0 (±5.0) | 86.7 (±2.9) | 83.3 (±5.8) | 96.7 (±2.9) | 85.0 (±5.0) |
| MVP | 96.7 (±2.9) | 98.3 (±2.9) | 81.7 (±2.9) | 91.7 (±2.9) | 86.7 (±2.9) | 20.0 (±5.0) | 65.0 (±8.7) | 75.0 (±8.7) | 96.7 (±2.9) | 76.7 (±11.6) |
| VC-1 | 98.3 (±2.9) | 98.3 (±2.9) | 78.3 (±2.9) | 86.7 (±2.9) | 95.0 (±5.0) | 21.7 (±2.9) | 66.7 (±2.9) | 86.7 (±2.9) | 98.3 (±2.9) | 71.7 (±2.9) |
| R3M | 91.7 (±2.9) | 71.7 (±16.1) | 21.7 (±2.9) | 63.3 (±5.8) | 36.7 (±2.9) | 35.0 (±8.7) | 76.7 (±2.9) | 43.3 (±7.6) | 71.7 (±2.9) | 58.3 (±5.8) |
| HRP | 98.3 (±2.9) | 98.3 (±2.9) | 90.0 (±0.0) | 65.0 (±0.0) | 96.7 (±2.9) | 23.3 (±2.9) | 61.7 (±2.9) | 85.0 (±0.0) | 96.7 (±2.9) | 81.7 (±2.9) |
| R3M-Droid | 98.3 (±2.9) | 96.7 (±5.8) | 90.0 (±0.0) | 80.0 (±0.0) | 83.3 (±5.8) | 38.3 (±2.9) | 66.7 (±2.9) | 61.7 (±20.2) | 98.3 (±2.9) | 83.3 (±5.8) |

### A.7 More baseline results on Robomimic

To further demonstrate the superior performance of RPM, we conduct more comparisons on Robomimic. We further expand our comparisons by including results from MAE (used in MVP and VC-1) pre-trained on DROID. Our analysis shows that while R3M benefits significantly from robotics data due to its Time Contrastive Learning that leverages trajectory information, MAE's performance degrades when trained on DROID due to reduced image quantity and lack of robotics-specific design. This highlights the importance of designing representation learning methods specifically for robotics datasets. We also compare with the multi-model representation LIV (Ma et al., 2023a) and standard vision baselines including ImageNet and SAM2. As shown in Table 7, RPM outperforms these baselines significantly.

### A.8 Comparison on visual generalization ability

We compare the generalization ability of some methods on Robomimic. While maintaining our original training protocol, we introduced color perturbations during policy evaluation by applying ColorJitter transformations to the visual observations.

**Table 7:** More baseline results in Robomimic.

| Task | LIV | ImageNet | SAM2 | MAE-DROID | **RPM** |
|---|---|---|---|---|---|
| Lift | 86.7 | 80.0 | 60.7 | 50.7 | **98.7** |
| Can | 4.0 | 1.3 | 5.3 | 4.0 | **68.3** |
| Square | 15.3 | 4.7 | 2.0 | 6.7 | **31.3** |
| All | 35.3 | 28.7 | 22.7 | 20.5 | **66.1** |

Specifically, our evaluation protocol involved selecting optimal model checkpoints from the training process, conducting 20 evaluation episodes (each with a duration of 200 timesteps), and recording success rates under standard conditions. We then repeated the evaluation with added visual perturbations using 'torchvision.transforms.ColorJitter(brightness=0.1, hue=0.1)'. The comparative results, shown in Table 8, demonstrate that our method exhibits superior robustness to visual perturbations, indicating stronger generalization capabilities across varying visual conditions.

**Table 8:** Comparison on visual generalization ability.

| | **RPM** | VC-1 | R3M |
|---|---|---|---|
| Original Performance | **95.0** | 83.0 | 86.7 |
| Performance after ColorJitter | **83.0** | 56.7 | 25.0 |
| Performance Drop Ratio (%, ↓) | **12.6** | 31.7 | 71.2 |

### A.9 Influence of each loss item on manipulation centricity

We also show analysis to reveal that each loss serves a distinct purpose: The dynamics alignment loss guides the encoder to capture rich manipulation-related features by focusing on the most dynamic elements of the scene. The action prediction loss directly enhances the model's action prediction capabilities by leveraging the pre-trained features. The time contrastive loss facilitates the learning of temporal visual relationships. Our experimental results on three challenging tasks, as shown in Table 9, confirm that all three losses contribute meaningfully to both improved manipulation centricity and higher downstream task success rates.

Moreover, we would like to emphasize two other key conclusions: (1) the dynamics alignment loss provides the most substantial benefit to representation learning, as it significantly enhances the encoder's focus on manipulation-centric features; and (2) the strong positive correlation between manipulation-centricity and downstream task success rates still hold, strengthening the key contribution of this paper.

**Table 9:** Influence of each loss item on manipulation centricity.

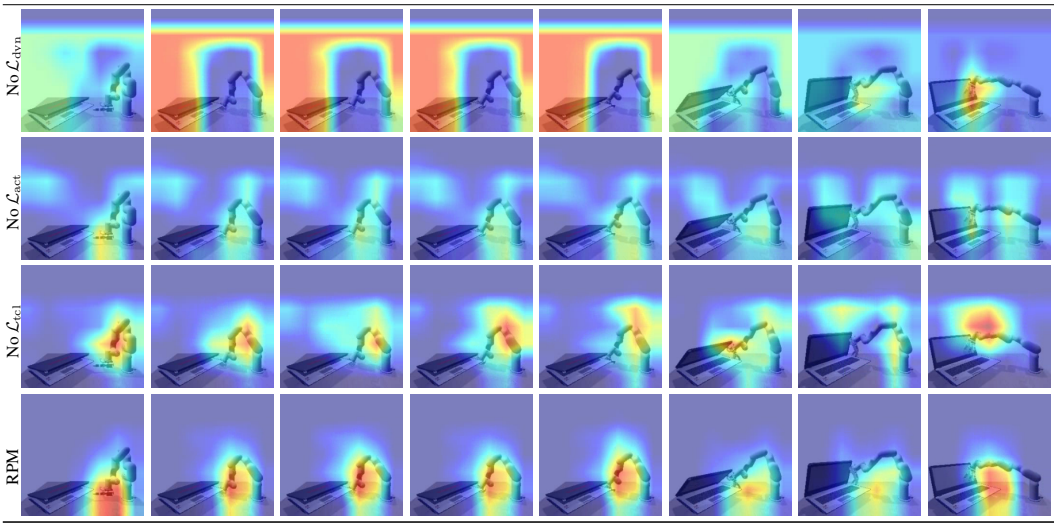
| Method | Manipulation Centricity | Success Rate |
|---|---|---|
| **RPM** | **0.175** | **83.2** |
| RPM w/o dynamics alignment | 0.100 | 66.2 |
| RPM w/o action prediction | 0.143 | 71.3 |
| RPM w/o temporal contrast | 0.145 | 72.0 |

We also provide qualitative results on how these losses impact manipulation centricity in Table 10. Removing time contrast and action prediction shows relatively smaller impacts on the Grad-CAM figures. The model can still focus on the manipulation-related regions, though not so well as RPM. However, when we remove dynamics alignment, the model fails to assign highlights to key regions, which leads to great degradation in both manipulation centricity and the success rate.

### A.10 Influence of camera parameters on manipulation centricity

The computation of manipulation centricity does not directly depend on the camera intrinsic or extrinsic parameters since it is derived from Grad-CAM videos and annotated segmentation videos, both of which are independent of specific camera parameters. Additionally, our evaluation suite deliberately includes diverse camera configurations to ensure robustness across different setups.

**Table 10:** Qualititive results on loss ablation.




While the absolute manipulation centricity values may vary with changes in camera intrinsics or extrinsics, the overall positive correlation between manipulation centricity and task success remains consistent. To demonstrate this, we conduct additional experiments using Robomimic demonstrations captured from an alternative camera viewpoint (eye-in-hand) and show the results in Table 11. Policies trained and evaluated under this configuration showed consistent trends.

**Table 11:** Influence of camera parameters on manipulation centricity in Robomimic.

| Method | Manipulation Centricity | Success Rate |
|---|---|---|
| VC-1 | 0.071 | 56.9 |
| R3M | 0.069 | 55.8 |
| **RPM** | **0.076** | **61.8** |

### A.11 Finetuning representation during policy training

We have also conducted additional experiments with full model finetuning for all methods, where our approach maintains its performance advantage while showing significant improvements over the frozen encoder results reported in the main paper, as illustrated in Table 12.

**Table 12:** Finetuning representation during policy training in Robomimic.

| Task | VC-1 | R3M | **RPM** |
|---|---|---|---|
| Can | 82.0 | 19.3 | **88.0** |
| Lift | 92.7 | 92.0 | **99.3** |
| Square | 38.0 | 27.0 | **56.0** |
| All | 70.9 | 46.1 | **81.1** |

### A.12 Quantitive results of Table 1.

To better clarify the results in Table 1, we calculate the manipulation centricities of the two tasks using the whole video. Results are shown in Table 13. Besides qualitative visualizations, it can be concluded from quantitative analysis that manipulation centricity is in positive correlation with the success rate.

## B More details on Manipulation Centricity

**Grad-CAM.** Grad-CAM is a widely used technique for generating heatmaps of input images to identify the regions that the encoder focuses on. We utilize the PyTorch-Grad-CAM library to

**Table 13:** Quantitive results of Table 1.

| task-metric | MVP | VC-1 | HRP | R3M | R3M-DROID | **RPM** |
|---|---|---|---|---|---|---|
| Square-manip. cen. | 0.025 | 0.060 | 0.059 | 0.057 | 0.062 | **0.078** |
| Square-success rate | 12.0 | 23.0 | 22.0 | 23.0 | 22.0 | **31.0** |
| P&P Wall-manip. cen. | 0.226 | 0.238 | 0.248 | 0.196 | 0.214 | **0.258** |
| P&P Wal-success rate | 86.0 | 96.0 | 95.0 | 36.0 | 83.0 | **100.0** |

generate Grad-CAM figures for each visual encoder. In convolutional neural networks (CNNs), Grad-CAM generates heatmaps by backpropagating gradients through the final convolutional layers, thereby highlighting important image regions. However, in Vision Transformers (ViT), the final classification is based on the class token computed in the last attention block, which means that the output is not influenced by the 14x14 spatial channels in the final layer, resulting in zero gradients. To generate meaningful visualizations in ViT, it is necessary to choose a layer before the final attention block. In our study, we chose the LayerNorm applied to the output of the self-attention mechanism in the last Transformer block.

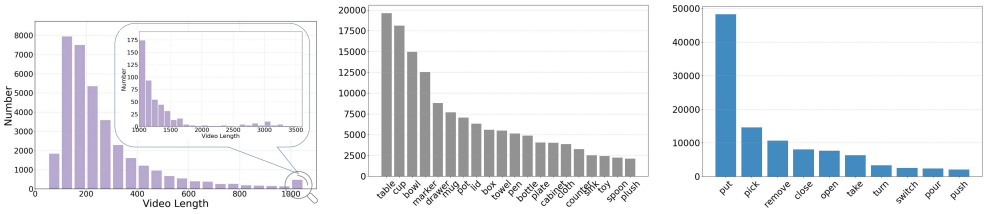
Our paper applies Grayscale CAM to measure Manipulation Centricity, where the pixel values range from 0 (black) to 255 (white), with higher values (closer to white) indicating regions that contribute more significantly to the model's decision. Furthermore, we binarize the Grayscale CAM by thresholding pixel values at 2, setting values below 2 to 0 and those above or equal to 2 to non-zero values, thereby mitigating the effect of noise and facilitating more accurate interpretations and precise metric calculations.

**SAM 2.** Segment Anything Model 2 (SAM 2, Ravi et al. (2024)) is a large-scale segmentation model. In our study, we utilize SAM2 to segment demonstration videos of all simulation tasks. Specifically, we select the official 'base plus' model available on the SAM2 codebase and adopt the implementation of the SAM2-GUI, an interactive graphical user interface to utilize pre-trained SAM2 models. We manually upload each task video and add prompt points to generate segmentation videos. Notably, foreground pixels, including robot end effectors and objects, are labeled with non-zero values, while background pixels are labeled with zero.

**Jaccard Index.** We employ the Jaccard Index to quantify the similarity between Grad-CAM and SAM2 video frames, with the index ranging from 0 to 1. A higher Jaccard Index value indicates a greater degree of similarity between the two, suggesting a stronger alignment between the attention maps generated by Grad-CAM and the segmentation masks produced by SAM2.

## C Detailed Analysis of DROID Subset Used

As described in Section 4.1, we perform a preliminary preprocessing of the full 1.7TB RLDS dataset, yielding a subset that serves as the basis for our analysis. In this section, we present a statistical characterization of this subset. Specifically, we exclude videos with fewer than 40 frames. The distribution of video lengths in the subset is illustrated in Figure 12a. Furthermore, we conduct a comprehensive statistical analysis of the language instructions accompanying the dataset, with a particular focus on the frequency of common object nouns and action verbs. The results of this analysis are visualized in Figures 12b and 12c, respectively. Notably, our subset covers a large degree of data diversity, capturing a wide range of scenarios, objects, and actions, thereby ensuring the effectiveness of our method.




**(a) Statistics of Trajectory Length.** **(b) Statistics of Common Object Noun.** **(c) Statistics of Action Verb.**

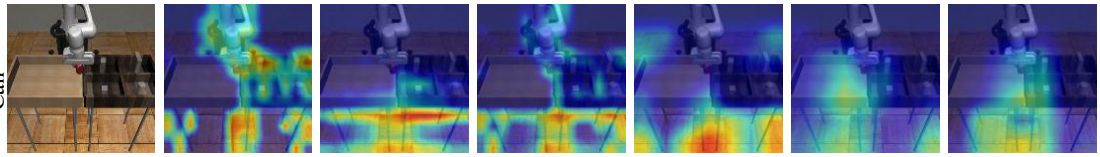
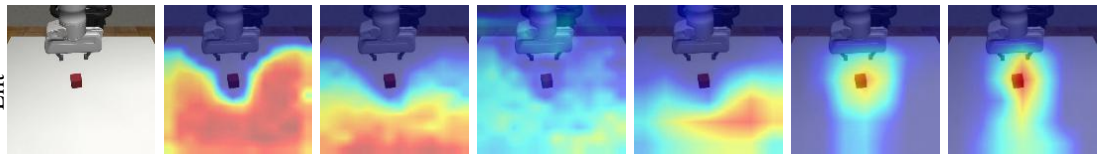
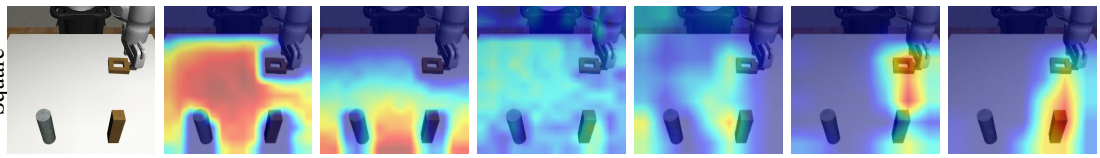
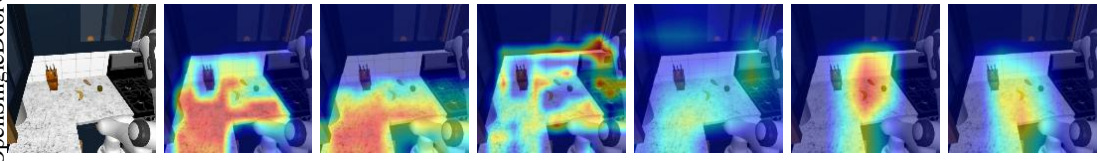
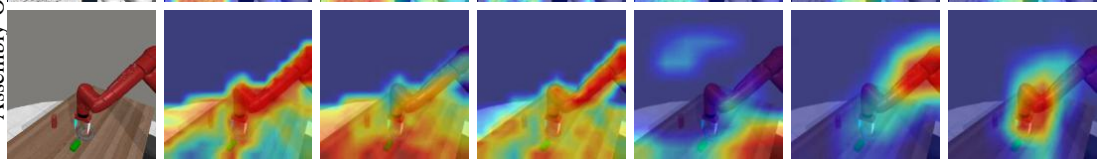
**Figure 12:** Statistical Analysis of the DROID Subset.

## D MORE RELATED WORK

Due to the space limit, we discuss more related work in this section.

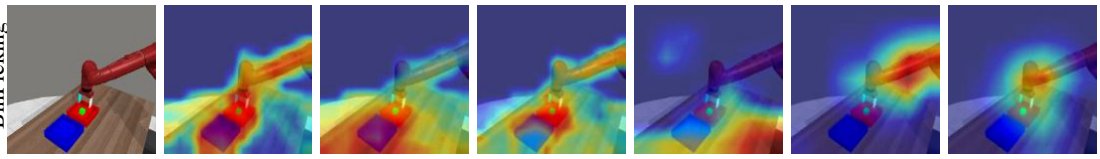
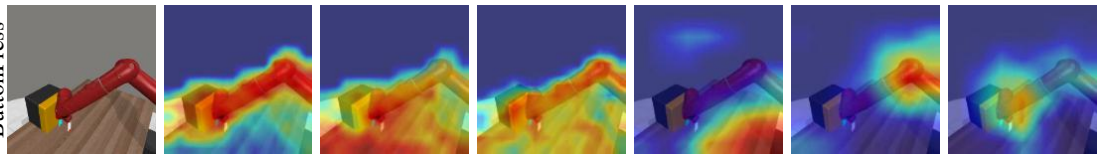
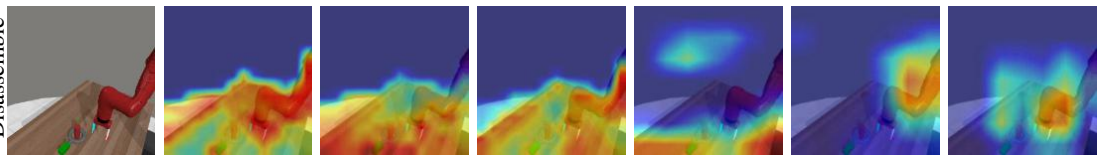
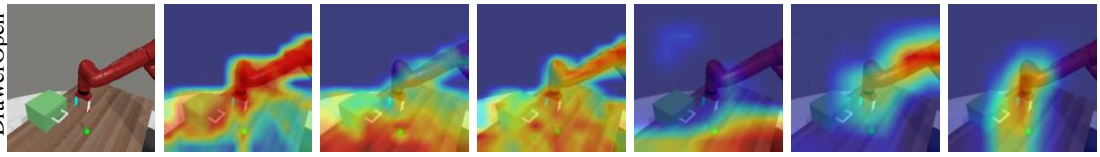
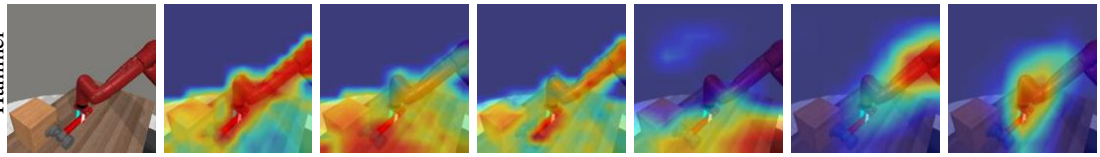
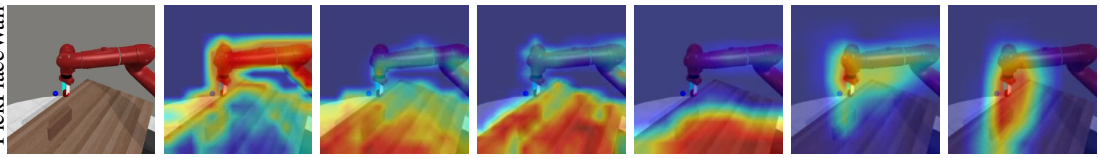
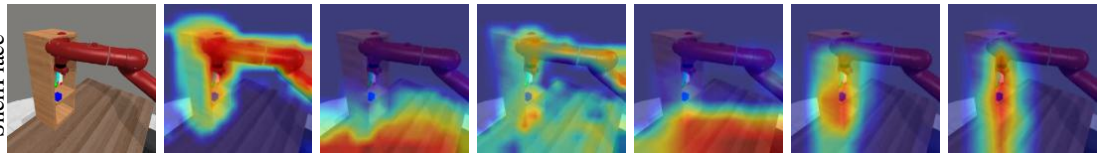
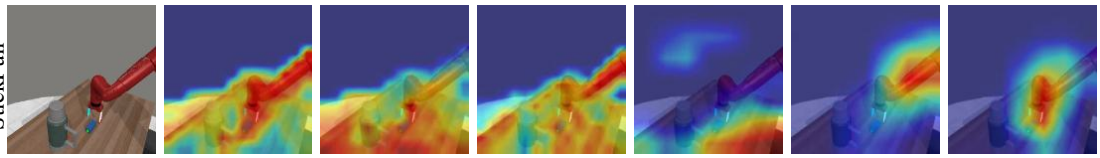
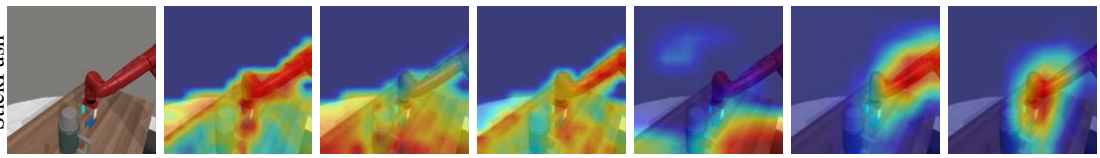
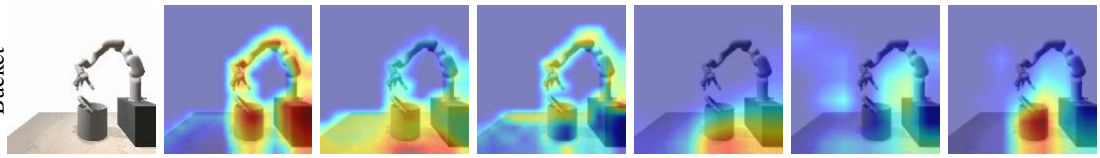
**Dynamics-aware representation.** Representation learning is crucial for extracting key features from image inputs in imitation learning and reinforcement learning Xu et al. (2023); Ji et al. (2024); Yuan et al. (2022). CURL (Laskin et al., 2020) uses InfoNCE (van den Oord et al., 2019) to maximize agreement between augmented observations, while CPC (Henaff, 2020) and ATC (Stooke et al., 2021) incorporate temporal dynamics into contrastive loss. DRIML (Mazoure et al., 2020) proposes a policy-dependent auxiliary objective, and KOROL (Chen et al., 2024) trains feature extractors on task-specific RGBD images during Koopman operator dynamics learning. TACO (Zheng et al., 2023; 2024; Liang et al., 2024) optimizes mutual information between current state-action pairs and future states. In contrast, our approach trains a generalized robotic representation model to extract more effective dynamics-aware representations, offering improved efficiency and generalization through a train-once-for-all methodology.

**Table 14: Grad-CAM of all tasks**

| Task | MVP | VC-1 | HRP | R3M | R3M-DROID | RPM |
|---|---|---|---|---|---|---|
| Can |  | | | | | |
| Lift |  | | | | | |
| Square |  | | | | | |
| CloseDrawer | | | | | | |
| PressButton | | | | | | |
| CoffeeDoor |  | | | | | |
| OpenSingle |  | | | | | |
| Assembly | | | | | | |

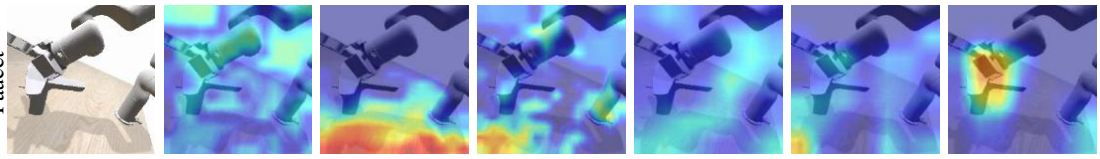
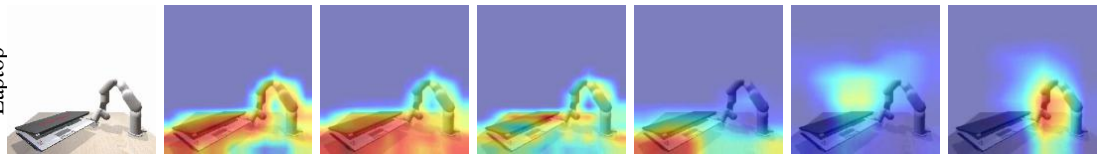
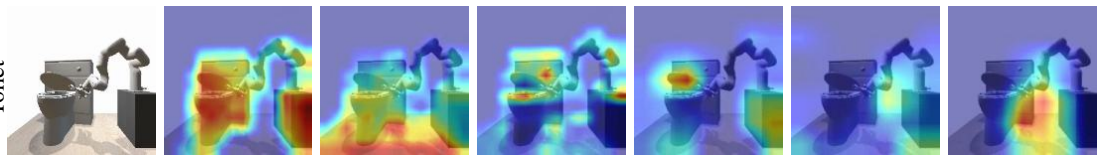
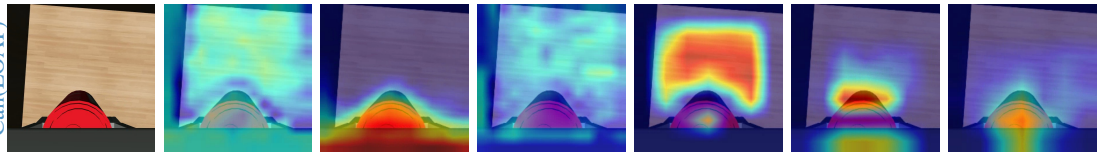
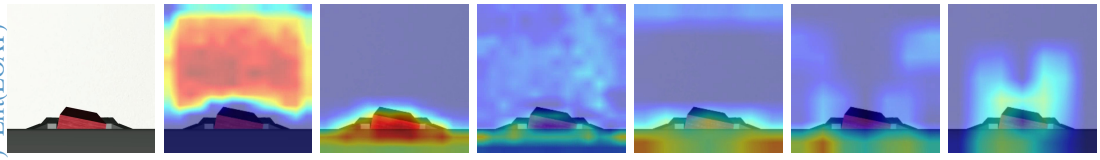
Continue

**Table 14: Grad-CAM of all tasks**

| Task | MVP | VC-1 | HRP | R3M | R3M-DROID | RPM |
|---|---|---|---|---|---|---|
| BinPicking |  | | | | | |
| ButtonPress |  | | | | | |
| Disassemble |  | | | | | |
| DrawerOpen |  | | | | | |
| Hammer |  | | | | | |
| PickPlaceWall |  | | | | | |
| ShelfPlace |  | | | | | |
| StickPull |  | | | | | |
| StickPush |  | | | | | |
| Bucket |  | | | | | |

Continue

**Table 14: Grad-CAM of all tasks**

| Task | MVP | VC-1 | HRP | R3M | R3M-DROID | RPM |
|---|---|---|---|---|---|---|
| Faucet |  | | | | | |
| Laptop |  | | | | | |
| Toilet |  | | | | | |
| Can(EOAT) |  | | | | | |
| Lift(EOAT) |  | | | | | |
| Square(EOAT) | | | | | | |