iManip: Skill-Incremental Learning for Robotic Manipulation

Zexin Zheng[1*], Jia-Feng Cai[1*], Xiao-Ming Wu[1], Yi-Lin Wei [1]
Yu-Ming Tang[1], Ancong Wu[1,2†], Wei-Shi Zheng[1,2†]
[1]School of Computer Science and Engineering, Sun Yat-sen University, China
[2]Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
{zhengzx25, caijf23}@mail2.sysu.edu.cn wuanc@mail.sysu.edu.cn wszheng@ieee.org

Abstract

The development of a generalist agent with adaptive multiple manipulation skills has been a long-standing goal in the robotics community. In this paper, we explore a crucial task, ***skill-incremental learning****, in robotic manipulation, which is to endow the robots with the ability to learn new manipulation skills based on the previous learned knowledge without re-training. First, we build a skill-incremental environment based on the RLBench benchmark, and explore how traditional incremental methods perform in this setting. We find that they suffer from severe catastrophic forgetting due to the previous methods on classification overlooking the characteristics of temporality and action complexity in robotic manipulation tasks. Towards this end, we propose an* ***i****ncremental* ***Manip****ulation framework, termed* ***iManip****, to mitigate the above issues. We firstly design a temporal replay strategy to maintain the integrity of old skills when learning new skill. Moreover, we propose the Extendable PerceiverIO, consisting of an action prompt with extendable weight to adapt to new action primitives in new skill. Extensive experiments show that our framework performs well in Skill-Incremental Learning.*

1. Introduction

Imagine that we are in a household setting with a robot assistant that already has basic functions like folding clothes and fetching items. Now, as the owners, we want it to learn new skills. For instance, today we've purchased a dishwasher, and we'd like the robot to learn how to load dishes into it. It could quickly acquire these new skills by observing and mimicking our demonstrations. This is an interesting and challenging requirement for robotics manipulation, which needs the robot to learn new skills based on previously learned knowledge without retraining. However, in the area of robotic manipulation, previous research mainly focuses on how to acquire better manipulation performance [6, 38, 47, 57, 69] or how to transfer the knowledge from the pretrained large language or vision models [4, 10, 21, 30] to learn robotics manipulations, rarely works explore how to incrementally learn new skills. LIBERO [35] is a preliminary benchmark to learn lifelong robotics control, but it explores the incremental abilities in new objects or new spatial positions, still limited in the characteristics of the same skills, where different tasks share the same skill.

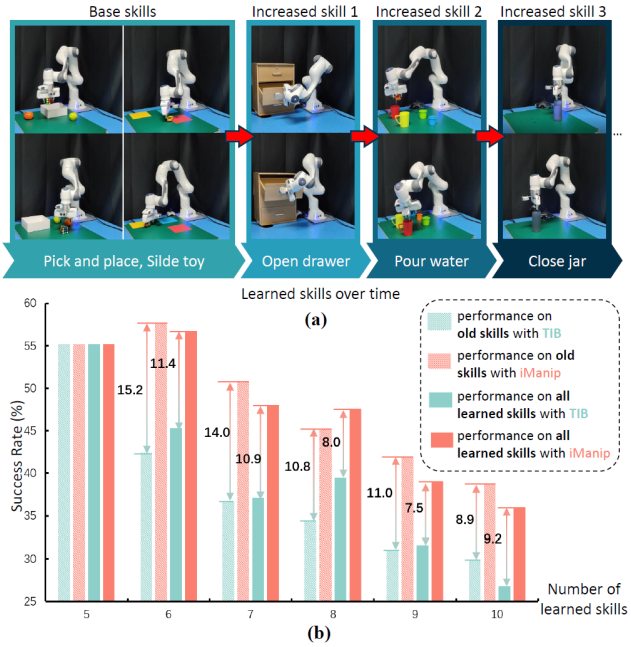

Figure 1. (a) An overview of our skill-incremental learning for robotic manipulation that requires the agent to learn skill sequences over time. (b) A comparison of model performance between the traditional incremental baseline (TIB) and our iManip.

In this paper, we thoroughly explore this crucial task, **skill-incremental learning**, in robotic manipulation, which is to leverage the previous learned knowledge to enable the robots to learn new manipulation skills without training from scratch. To begin with, we construct a skill-

*Equal contribution.
†Corresponding author.

incremental environment based on the RLBench benchmark. It requires the agent to continuously acquire a sequence of 10 challenging language-conditioned manipulation skills. Each skill consists of at least two variations encompassing several types, such as variations in shape and color, totaling 166 variations. Then, we explore how the traditional incremental learning methods work on this skill-incremental environment. As seen in Figure 1, we discover that after learning new manipulation skills, the agent's performance on prior skills significantly deteriorates. Thus, we can conclude that traditional incremental learning methods still suffer from catastrophic forgetting in this new setting, which, we regard, is due to their neglect of the temporality and action complexity in robotic manipulation tasks. The temporal complexity arises from the dynamic changes in the environment and in the robot states over time, resulting in each action having an impact on subsequent actions. And the complexity of actions requires the agent to learn new action primitives for action planning in novel environments and interactions, which is representative in 3D dimensions with rotation and shift, and highly complex.

To mitigate the above problems, we propose a new skill **i**ncremental **Manip**ulation framework, termed **iManip**, for this new setting. The key idea of our framework is to modify the traditional incremental learning methods to fully consider the characteristics of temporality and action complexity in robotic manipulation tasks. Two designs make our framework nontrivial. First, to address temporal complexity, we design the temporal replay strategy to maintain the integrity of the temporal data and propose to replay a fixed number of keyframe samples at different time points for each manipulation skill, using the farthest-distance entropy sampling strategy. Second, to address the complexity of actions, we propose the Extendable PerceiverIO, consisting of an action prompt with extendable weight to adapt to new action primitives. When learning new skills, we freeze the learned parameters of PerceiverIO while learning a new small set of skill-specific action prompts and weight matrices for new action primitives learning.

Extensive experiments show that our iManip framework maintains several excellent capabilities: (1) **Effectiveness**: it performs well in the skill-incremental learning setting, outperforming the traditional incremental baseline with an increase of **9.4** points. (2) **Robustness**: it demonstrates robust performance in several different incremental settings. (3) **Lightweight**: it only needs lightweight finetuning of the policy decoder with fewer training steps, comparable with full weights finetuning. (4) **Extendability**: it also has extraordinary performance in real-world experiments.

2. Related Work

Robotic Manipulation. Learning robot manipulation conditioned on both vision and language has gained increasing attention [11, 16–18, 49, 50, 53, 58–60, 63, 65], with robot imitation learning using scripted trajectories [23, 39] or tele-operation data [12, 42, 62] gradually becoming a mainstream approach. Previous work [5, 13, 24] has focused on using 2D images to predict actions, while more recent studies have leveraged the rich spatial information of 3D point clouds for motion planning. For instance, PerAct [51] feeds voxel tokens into a PerceiverIO [22]-based transformer policy, achieving impressive results across various tasks. GNFactor [67] optimizes a generalizable neural field for semantic extraction, while ManiGaussian [37] introduces a dynamic Gaussian Splatting [27] framework for semantic propagation. 3DDA [26] proposes a 3D denoising transformer to predict noise in noised 3D robot pose trajectories. However, these methods suffer from catastrophic forgetting in skill-incremental learning and we propose our iManip framework to continuously learn new skills and mitigate forgetting of learned knowledge.

Conventional Lifelong Learning Approaches. One of the most commonly used methods is the rehearsal-based method [2, 8, 9, 19, 20, 25, 33, 46, 54, 56]. iCaRL [46] proposes a herding-based step for prioritized exemplar selection to store old exemplars. RWalk [8] proposes a hard-exemplar sampling strategy for replay. Distillation-based methods [7, 14, 19, 31, 34, 52, 55, 68] propose distilling old knowledge from the old network to the current network or maintaining the old feature space during new tasks. ABD [52] proposes distilling synthetic data for incremental learning and EEIL [7] proposes to distill the knowledge from the classification layers of the old classes. Moreover, there are dynamic-architecture-based methods [1, 32, 36, 43, 44, 66] that dynamically adjust the model's representation ability to fit the evolving data stream. In this work, we use the traditional rehearsal-based method [46] and the distillation-based methods [7] for robotic skill-incremental learning. We find that they also suffer from catastrophic forgetting due to overlooking the temporal and action complexities of robotic manipulation.

A Preliminary Lifelong Robot Learning Benchmark LIBERO. Previous works [15, 28, 29, 40, 41, 61] explore different strategies for incremental robotic learning based on different testing environments. Recently, to promote community development, LIBERO [35] proposed a benchmark, which explores incremental abilities with new objects, goals, or spatial positions, as shown in Figure 2. The limitation of LIBERO lies in the fact that most tasks are constrained by similar skill characteristics. For example, it regards "Put the bowl on the plate" and "Put the bowl on stove" as different tasks. In this paper, we explore a more realistic and challenging task, skill-incremental learning, where the agent learns a sequence of skills over time, each involving multiple poses and object variations in placement, color, shape, size, and category.

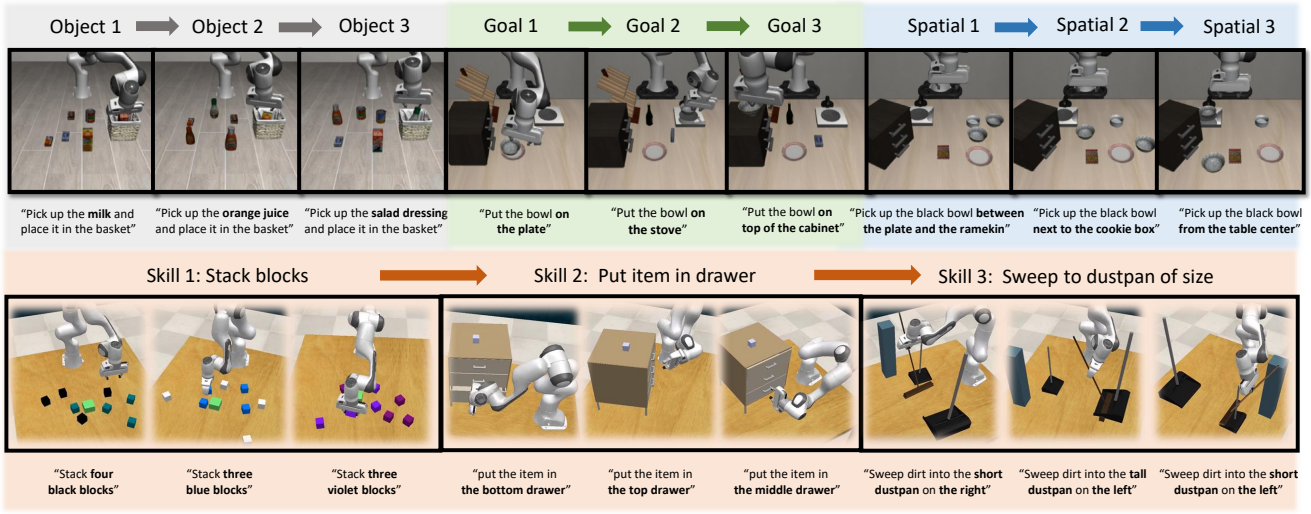

Figure 2. Overview of robotic incremental learning. Previous works focus on incremental abilities in new objects, goals, or spatial positions, where different tasks may share the same skill. The iManip focuses on skill-incremental learning which better captures the true adaptability and flexibility required for real-world robotic learning.

3. Skill-incremental Learning for Robotic Manipulation

3.1. Challenges

While robotic manipulation has received increasing attention, few previous studies explore how to incrementally learn new skills. In this paper, we focus on skill-incremental learning for robotic manipulation, a challenging setting that requires agents to continuously acquire new skills without training from scratch.

In this new setting, we find that applying traditional incremental learning methods [7, 8, 37, 46, 51] still suffers from catastrophic forgetting of previously learned skills, as seen in Figure 1 (b). There are two key challenges when applying previous methods of visual classification to robotic skill-incremental learning: 1) Previous methods overlook the temporal complexity inherent in robotic manipulation tasks, where dynamic changes in the environment and the robot states over time cause actions to impact subsequent ones. For example, classical replay algorithms focus on sampling the most representative samples per class, directly storing representative samples from demonstrations may result in temporal imbalance of the trajectory, leading to instability during task execution. 2) Previous methods focus primarily on general visual features while neglecting the actions complexity in robotic manipulation. Robotic manipulation involves action planning through interactions with the physical environment, such as visual and language input. When a new manipulation skill arises, the agent learns new visual-language interactions and quickly acquires new action primitives based on prior knowledge.

3.2. Overall Pipeline

To tackle the two challenges, we propose the temporal replay strategy and the extendable PerceiverIO architecture in our iManip framework. Specifically, as shown in Figure 3, we present the overall framework for robotic skill-incremental learning, which can sequentially learn robotic manipulation skills while mitigating catastrophic forgetting of learned skills. Specifically, the agent learns a sequence of manipulation skills with a stream of training data denoted as $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^{T}$, where $\mathcal{D}_i = \{(o_i^{(1)}, a_i^{(1)}), (o_i^{(2)}, a_i^{(2)}), \dots\}$ represents the demonstration trajectories of skill i. The visual input $o_i^{(t)} = (I_i^{(t)}, D_i^{(t)}, P_i^{(t)})$ consists of the t-th single-view images $I_i^{(t)}$, depth images $D_i^{(t)}$, and proprioception matrix $P_i^{(t)} \in \mathbb{R}^4$ that includes the openness, end-effector position, and the current timestep. After learning skill i, a compact memory $\mathcal{M}$ stores a fixed number of demonstration replays for skills up to $i-1$. Following [37, 51, 67], the agent combines the visual input $o_i^{(t)}$ and language instructions l_i to generates the optimal action $a_i^{(t)} = (a_{i,\text{trans}}^{(t)}, a_{i,\text{rot}}^{(t)}, a_{i,\text{open}}^{(t)}, a_{i,\text{col}}^{(t)})$, which respectively demonstrates the target translation in voxel $a_{i,\text{trans}}^{(t)} \in \mathbb{R}^{100^3}$, rotation $a_{i,\text{rot}}^{(t)} \in \mathbb{R}^{(360/5)\times 3}$, openness $a_{i,\text{open}}^{(t)} \in [0,1]$ and collision avoidance $a_{i,\text{col}}^{(t)} \in [0,1]$.

Our framework consists of a voxel encoder for learning 3D scene features, a latent transformer (the extendable PerceiverIO), and a policy decoder to predict optimal robot actions. Specifically, our approach employs a temporal replay strategy, maximizing the information entropy of replay demos to effectively address the first challenges caused by

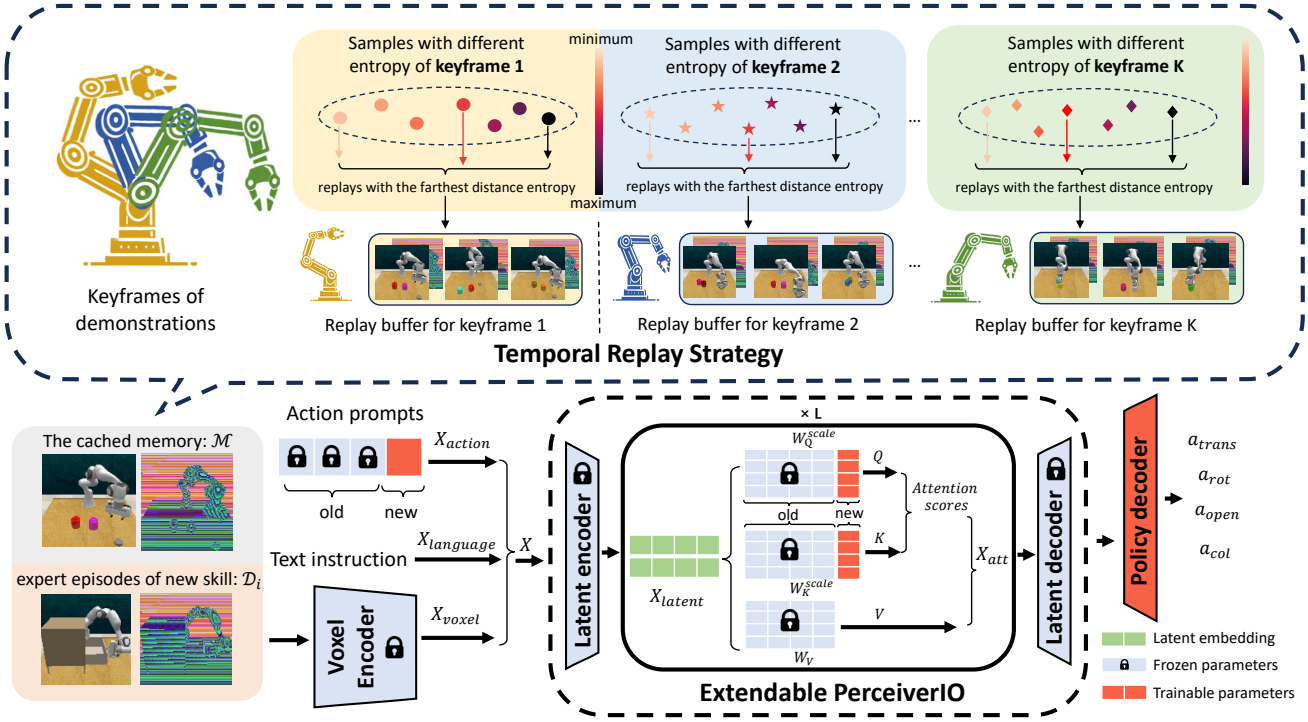

Figure 3. The overall framework of iManip, which primarily consists of a temporal replay strategy to store the samples with the farthest distance entropy for each keyframe of old demonstrations and an extendable PerceiverIO consisting of action prompts with extendable weights to adapt to new action primitives.

classic replay methods. Additionally, we introduce the extendable PerceiverIO, consisting of action prompts with extendable weights to adapt to new action primitives, while preserving knowledge of previous skills to prevent catastrophic forgetting.

3.3. The iManip Framework

Temporal Replay Strategy. Rehearsal-based methods [8, 19, 46] are classic algorithms in incremental vision classification tasks while overlooking the temporal complexity inherent in robotic manipulation. Apart from random sampling, popular methods such as herding sampling [46] and hard-exemplar sampling [3, 8] effectively select the most representative samples from each class. However, robotic manipulation data consists of temporal samples from entire expert episodes. Directly storing representative samples can lead to temporal imbalance of episodes, resulting in instability during task execution.

Therefore, we propose a temporal replay strategy to balance the sampling of different keyframes of episodes for each skill. Keyframes [51] are the samples from episodes when the end-effector changes state (e.g., gripper closing) or when its velocity approaches zero, representing critical temporal landmarks within the trajectory.

Furthermore, to sample a greater variety of variants, we propose the farthest-distance entropy sampling to store an equal number of each keyframe. It requires the buffer to contain the replays that exhibit the largest entropy divergence as

$$S = \underset{S \in E, |S|=K}{\arg\max} \sum_{i \in S} \sum_{j \in S} A[i][j], \tag{1}$$

where S is the sampled set with size K, E is the demo set of a specific keyframe with size N, and A is the distance array of the demos' action prediction entropy $\mathcal{L}_{\text{act}}$. Specifically, we propose to store the demo j to the replay buffer that has the farthest distance of sampled demos in S as

$$j = \underset{j \in E}{\arg\max} \sum_{k \in S} A[j][k]. \tag{2}$$

This ensures the storage of temporally balanced, information-rich samples from previous episodes, helping to mitigate catastrophic forgetting of learned skills.

Based on the above analysis, the pseudocode for the temporal replay strategy is shown in Algorithm 1. The algorithm can get the optimal solution for the objective 1, with time complexity of $O(N^2)$, which is the size of a specific keyframe and not relevant to the size of the previous data, suitable for incremental skill learning.

Algorithm 1 Farthest-distance Entropy Sampling

Require: Entropy set corresponding to the keyframe samples $E = \{e_1, e_2, \ldots, e_N\}$, sampling size K
Ensure: Sampled set $S = \{s_1, s_2, \ldots, s_K\}$
1: Calculate the distance array A, where $A[i][j] = \text{distance}(e_i, e_j)$
2: Select the sample i and add to S, where $i = \arg\max_{1 \le i \le N} \sum_{j=1}^{N} A[i][j]$
3: **for** $k = 2$ to K **do**
4: Find the value in E that has the largest difference with the values in S, $j = \arg\max_{j \in E} \sum_{k \in S} A[j][k]$
5: Add sample j to S as the k-th sample
6: **end for**
7: **return** Sampled set S

Extendable PerceiverIO. Unlike traditional vision classification tasks, robotic manipulation requires the integration of multiple modalities to enable complex decision-making and long-term action planning through interactions with the physical environment. Classic methods [7, 8, 19, 46] for lifelong classification overlook the complexities of action in robotic manipulation tasks that require the agent to learn various action primitives for different skills. In our iManip framework, we propose an extendable PerceiverIO, which learns skill-specific action prompts with extendable weights to adapt to new action primitives.

Specifically, as illustrated in Figure 3, the input to the extendable PerceiverIO consists of multimodal patches, $X = [X_{\text{voxel}}, X_{\text{language}}, X_{\text{action}}]$, where X_{voxel} and X_{language} represent the input sequences of voxel and language encodings, respectively, and $X_{\text{action}} = [X^{\text{old}}_{\text{action}}, X^{\text{new}}_{\text{action}}]$ is the skill-specific action prompt that concatenates both the old and new action prompts. The notation $[\cdot, \cdot]$ denotes the concatenation operation along the token dimension. Subsequently, X undergoes a cross-attention computation between the input and a much smaller set of latent vectors through the latent encoder, producing X_{latent}. Then X_{latent} is encoded with weight-extendable self-attention layers as

$$Q = X_{\text{latent}} \cdot W^{\text{scale}}_{Q}, \quad K = X_{\text{latent}} \cdot W^{\text{scale}}_{K}, \quad V = X_{\text{latent}} \cdot W_{V}, \tag{3}$$

$$X_{\text{att}} = \text{softmax}[\frac{Q \cdot K^{\top}}{\sqrt{d}}] \cdot V, \tag{4}$$

where $X_{\text{latent}}, X_{\text{att}} \in \mathbb{R}^{T \times d}$ are respectively a set of T input and output tokens with channel dimension d, $W^{\text{scale}}_{Q}, W^{\text{scale}}_{K} \in \mathbb{R}^{d \times d'}$,$W_V \in \mathbb{R}^{d \times d}$ are learnable weight matrices. Notably, $W^{\text{scale}}_{Q}, W^{\text{scale}}_{K}$ are extendable by appending newly weight matrices $W^{\text{new}}_{Q}, W^{\text{new}}_{K} \in \mathbb{R}^{d \times d_{\text{new}}}$ as

$$W^{\text{scale}}_{Q} = \left[W^{\text{old}}_{Q}, W^{\text{new}}_{Q}\right], \quad W^{\text{scale}}_{K} = \left[W^{\text{old}}_{K}, W^{\text{new}}_{K}\right], \tag{5}$$

where $W^{\text{old}}_{Q}, W^{\text{old}}_{K} \in \mathbb{R}^{d \times d_{\text{old}}}$ are the old weight matrices and the expanded dimension $d' = d_{\text{old}} + d_{\text{new}}$. Finally, these encoded latents are cross-attended with the input once again through the latent decoder to ensure alignment with the input size. In our iManip framework, we freeze the old PerceiverIO while learning the action prompts $X^{\text{new}}_{\text{action}}$ and a small set of newly weight matrices $W^{\text{new}}_{Q}, W^{\text{new}}_{K}$ of new skills. This enables the agent to quickly adapt to new action primitives while preventing the forgetting of previous skills.

Knowledge distillation between the old and new agents. To better preserve the knowledge of previous skills while learning new ones, we employ knowledge distillation [7, 64], where the output probability distribution of the old model is used to train the new model. This enables the transfer of knowledge from the old to the new agent, as defined by the following objective:

$$\begin{aligned} \mathcal{L}_{\text{dis}} = \mathcal{L}_2(\mathcal{Q}^{\text{old}}_{\text{trans}}, \mathcal{Q}^{\text{new}}_{\text{trans}}) + \mathcal{L}_2(\mathcal{Q}^{\text{old}}_{\text{rot}}, \mathcal{Q}^{\text{new}}_{\text{rot}}) + \\ |\mathcal{Q}^{\text{old}}_{\text{open}} - \mathcal{Q}^{\text{new}}_{\text{open}}| + |\mathcal{Q}^{\text{old}}_{\text{collide}} - \mathcal{Q}^{\text{new}}_{\text{collide}}|, \end{aligned} \tag{6}$$

where $\mathcal{L}_2$ is the MSE loss, $[\mathcal{Q}^{\text{old}}_{\text{trans}}, \mathcal{Q}^{\text{old}}_{\text{rot}}, \mathcal{Q}^{\text{old}}_{\text{open}}, \mathcal{Q}^{\text{old}}_{\text{collide}}]$ and $[\mathcal{Q}^{\text{new}}_{\text{trans}}, \mathcal{Q}^{\text{new}}_{\text{rot}}, \mathcal{Q}^{\text{new}}_{\text{open}}, \mathcal{Q}^{\text{new}}_{\text{collide}}]$ denote the probabilities of the ground truth actions in expert demonstrations for translation, rotation, gripper openness, and collision avoidance for the old and new robots, respectively.

3.4. Learning Objectives

Our approach is to address the problem of skill-incremental learning for robotic manipulation from multiple aspects. First, for each manipulation skill, there is an action loss to facilitate robot imitation learning. Following [37, 51, 67], we employ cross-entropy loss (CE) to ensure accurate action prediction:

$$\begin{aligned} \mathcal{L}_{\text{act}} = -\mathbb{E}_{Y_{\text{trans}}}\left[\log \mathcal{V}_{\text{trans}}\right] - \mathbb{E}_{Y_{\text{rot}}}\left[\log \mathcal{V}_{\text{rot}}\right] \\ -\mathbb{E}_{Y_{\text{open}}}\left[\log \mathcal{V}_{\text{open}}\right] - \mathbb{E}_{Y_{\text{collide}}}\left[\log \mathcal{V}_{\text{collide}}\right], \end{aligned} \tag{7}$$

where $\mathcal{V}_i = \text{softmax}(\mathcal{Q}_i)$ for $\mathcal{Q}_i \in [\mathcal{Q}_{\text{trans}}, \mathcal{Q}_{\text{open}}, \mathcal{Q}_{\text{rot}}, \mathcal{Q}_{\text{collide}}]$ and $Y_i \in [Y_{\text{trans}}, Y_{\text{rot}}, Y_{\text{open}}, Y_{\text{collide}}]$ is the ground truth one-hot encoding.

Furthermore, when learning new skills, we propose the temporal replay strategy to preserve a fixed number of representative samples from old demonstrations. The cached memory $\mathcal{M}$ will be used in conjunction with the new skill demos $\mathcal{D}_{\text{new}}$ for learning new skills. Additionally, our extendable PerceiverIO will dynamically expand new learnable weights for the new skill. We find that training only the skill-specific action prompts $X^{\text{new}}_{\text{action}}$ with newly appended weights $W^{\text{new}}_{Q}, W^{\text{new}}_{K}$ and the policy decoder effectively prevents catastrophic forgetting, more analysis can be seen in the 3rd experiments in Section 4.2. Finally, we employ knowledge distillation loss $\mathcal{L}_{\text{dis}}$ to help the agent retain the knowledge of previous skills. Overall, in skill-incremental

Methods	Base	Step 1		Step 2		Step 3		Step 4		Step 5		Average	
		Old	All	Old	All	Old	All	Old	All	Old	All	Old	All
multi-task methods													
PerAct [51]	44.0	4.0	7.3	2.7	5.1	1.1	9.0	2.5	6.7	1.3	1.6	2.3	5.9
ManiGaussian [37]	55.2	12.0	20.7	6.7	12.0	5.7	15.5	3.0	9.3	5.3	5.2	6.5	12.5
skill-incremental methods													
P-TIB [7, 46, 51]	44.0	33.6	34.7	26.0	25.1	22.3	26.0	17.0	16.4	11.6	10.4	22.1	22.5
M-TIB [7, 37, 46]	55.2	42.4	45.3	36.7	37.1	34.3	39.5	31.0	31.6	29.8	26.8	34.8	36.1
Ours (iManip)	**56.0**	**57.6**	**56.7**	**50.7**	**48.0**	**45.1**	**47.5**	**42.0**	**39.1**	**38.7**	**36.0**	**46.8**	**45.5**

Table 1. Performance comparison of different methods of B5-5N1 in Rlbench. We show the average success rate of old and all learned skills, and the average performance of all new steps. The traditional incremental methods [7, 46] on baseline [37, 51] is termed TIB.

learning, our training loss is formulated as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{act}} + \lambda_{\text{dis}}\mathcal{L}_{\text{dis}}, \tag{8}$$

where λ_{dis} is a hyperparameter that controls the importance of the knowledge distillation loss $\mathcal{L}_{\text{dis}}$ during training.

4. Experiments

4.1. Experimental setup and details

Experimental setup. Following [37, 67], we select 10 representative manipulation skills in RLBench[23] and 5 daily manipulation skills in the real world for our experiments. Each skill has at least two variations and 20 demonstrations during training that cover multiple types, such as position, shape, and color. To achieve a high success rate for these skills, the manipulation policy needs to learn generalizable knowledge rather than overfitting the limited given demonstrations. For visual observation, we only use the front RGB-D image with 128×128 resolutions. To demonstrate the performance of our method under different incremental settings, we define several configurations, represented as **Bn-kNm**. This notation indicates that the policy is initially trained on **n** base skills, followed by the addition of **m** new skills in each step, with a total of **k** steps.

Evaluation Metric. We report the performance of the agent on each learned skill by the average success rate. At each incremental step, we present the average success rate for old, new, and all (combined old and new) skills. In the simulation, we evaluate the agent with 25 episodes per skill, whereas in the real world, we use 10 episodes per skill. During evaluation, the agent continues to take actions until an oracle signals task completion or the agent reaches a maximum of 25 steps.

Implementation Details. For model design, we use different encoders to transform corresponding modality data into tokens, which serve as the input for the Extendable PerceiverIO. The RGB-D images are projected and transformed into voxels, which are then encoded by a 3D convolutional encoder with a UNet architecture, while text instructions are encoded using CLIP RN50 [45], and the proprioception data is encoded by a single-layer MLP. After encoding, the tokens from all modalities have the same dimension of 512. The hyperparameter λ_{dis} is set as 0.01 and the action prompt length is 16. We store 2 keyframe replays of the total 20 demonstrations of learned skills and train the agent on two NVIDIA RTX 4090 GPUs with a batch size of 1, a learning rate of 0.002, and 100k iterations. More studies about the hyperparameters are shown in the Appendix.

	TRS	EPIO	DIS	B5-1N1	B5-5N1
R1				20.7	5.2
R2	✓			49.3	27.6
R3	✓	✓		54.0	32.4
Ours	✓	✓	✓	56.7	36.0

Table 2. Ablation Study on two experiment setup. We report the average success rate of all learned skills.

4.2. Simulation results

Performance comparison with different methods. We conduct the skill-incremental learning experiment in the B5-5N1 setting, where we first train the policy on five base skills and then gradually learn a new skill at each subsequent step with a total of five steps.

To demonstrate the performance of our method in robotic skill-incremental learning, we compare with two standard multi-task manipulation policies, PerAct [51] and ManiGaussian [37], by retraining the agent to learn new manipulation skills. Furthermore, we apply two Traditional Incremental Baselines (TIB) for visual classification to the above policies for comparison, termed P-TIB and M-TIB respectively. As shown in Table 1, the results demonstrate that the performance of our method significantly outperforms the others at each subsequent step. This demonstrates that our method better facilitates the learning of new skills while mitigating the forgetting of previous skills. More detailed results of each learning skill at every step are shown in the Appendix.

Ablation Study. We conduct the ablation study to validate the effectiveness of each policy, as shown in Table 2. R1 is the control group where the agent does not have any incremental policy. When we add the Temporal Replay Strategy

Frozen layer	Convergence steps	Trained param	Slide block		Put in drawer		Drag stick		Push buttons		Stack blocks	
			Old	New	Old	New	Old	New	Old	New	Old	New
Non-frozen	100000	47M	43.2	**60.0**	44.8	**16.0**	40.8	**92.0**	44.0	**28.0**	45.6	**12.0**
Encoder	75000	37M	50.4	54.0	51.2	12.0	45.6	88.0	48.4	24.0	52..0	8.0
EPIO	70000	18M	52.0	52.0	52.8	16.0	46.4	84.0	50.4	24.0	49.6	8.0
Decoder	75000	39M	45.6	24.0	47.2	0.0	42.4	40.0	44.8	4.0	46.4	0.0
Encoder+EPIO	60000	8M	**57.6**	52.0	**56.8**	12.0	**50.4**	84.0	**55.2**	20.0	**56.0**	8.0

Table 3. Performance of five sets B5-1N1 experiments with the same base skills and different new skills, while freezing different network layers. For each new skill, we train a total of 100k iterations and report the average success rate and the average model convergence steps.

Methods	B5-1N5	B2-4N2	B3-2N3
ManiGaussian [37]	25.6	10.4	17.3
M-TIB [7, 37, 46]	30.8	28.4	33.3
Ours (iManip)	**37.2**	**36.8**	**41.3**

Table 4. Average success rate of all learned skills on different skill-incremental setup.

(**TRS**) to the agent, the setup of B5-1N1 and B5-5N1 improve by 28.6% and 22.4% in the success rate, respectively (see R2). The significant performance improvement stems from the success of our temporal replay strategy that maintains the integrity of the temporal data.

When we add Extendable PerceiverIO (**EPIO**) to the agent, the success rates of B5-1N1 and B5-5N1 further improve by 4.7% and 4.8%, respectively (see R3). The EPIO design works because the skill-specific action prompts help the agents incrementally learn action primitives for new skills, and the extendable weights designed in transformer blocks allow the model to preserve the old knowledge while adapting to new skills. Our complete policy achieves the best success rate, where the Distillation mechanism (**DIS**) improves the performance by 2.7% and 3.6%, respectively.

Through the ablation study, we find that the replay policy has the greatest impact on overall performance. Without old data for retraining, the agent is more likely to forget previously learned knowledge. This occurs because data plays a crucial role in robotic manipulation, and without the support of previous data, the agent is prone to overfitting the data of new skills.

Effect of parameter freezing on skill-incremental learning. The agent consists of three main components: the encoders, the extendable PerceiverIO, and the policy decoder. we freeze each component individually to evaluate its effect. As shown in Table 3, five sets B5-1N1 experiments on different new skills demonstrate the following: (1) Freezing the encoders or the extendable PerceiverIO helps retain knowledge from the old skills without significantly hindering the learning of the new skill (Lines 1,2,3). (2) The policy decoder is crucial for learning new skills (Lines 1,4). (3) Freezing the above components helps decrease the number of parameters and accelerate convergence.

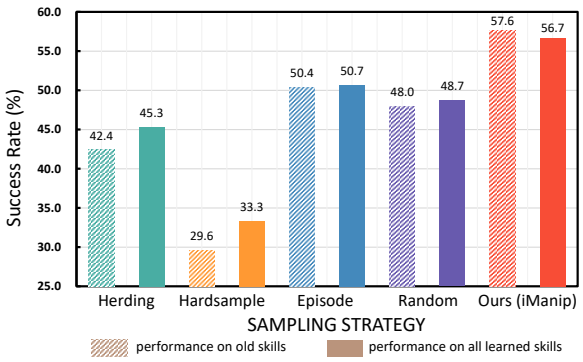

Figure 4. Average success rate on different replay methods.

Based on these findings, we freeze both the encoder and the extendable PerceiverIO, leaving only the decoder with newly appended action prompts and weights for new skill training, achieving faster convergence, fewer parameters, and better performance (Line 5)!

Experiments on different skill-incremental setup. We implement different skill-incremental settings to validate the generalizability of our method and report the average success rate across all previously learned skills after the last incremental step. As shown in Table 4, compared to ManiGaussian and M-TIB, our method achieves higher success rates in all incremental experimental settings. This shows that our approach has stronger performance and better generalization capabilities.

Exploring data replay methods. We compare our temporal replay strategy with the classical rehearsal-based method on the setup of B5-1N1, as shown in Figure 4. **Herding** [46] and **Hardsample** [8] are two methods for selecting the most representative samples from the data. **Episode** refers to replaying a complete trajectory. **Random** refers to random sampling. The results show that classic herding sampling and hard-exemplar sampling perform poorly on old skills due to neglecting temporal integrity in robotic demonstrations. In contrast, replaying complete trajectories or random sampling better preserves the temporal integrity of the samples, leading to better performance. Our temporal replay strategy, leveraging the farthest-distance entropy sampling for each keyframe can sample more different variants and achieves the best success rate.

Manipulation skills	Base		Step 1		Step 2		Step 3		Step 4	
	BL	Ours	BL	Ours	BL	Ours	BL	Ours	BL	Ours
Slide toy	90.0	90.0	10.0	80.0	0	80.0	0	60.0	0	60.0
Open drawer	-	-	70.0	60.0	0	60.0	0	50.0	0	40.0
Pick and place	-	-	-	-	60.0	60.0	0	60.0	0	50.0
Pour water	-	-	-	-	-	-	40.0	40.0	0	10.0
Close jar	-	-	-	-	-	-	-	-	50.0	40.0
Old manipulation skills	90.0	90.0	10.0	**80.0** +70.0	0	**70.0** +70.0	0	**56.7** +56.7	0	**40.0** +40.0
All manipulation skills	90.0	90.0	40.0	**70.0** +30.0	20.0	**66.7** +46.7	10.0	**52.5** +42.5	10.0	**40.0** +30.0

Table 5. Real world experiments. The table reports the success rate of BaseLine (BL) and Ours. +num is the improvement of our method compared to the baseline.

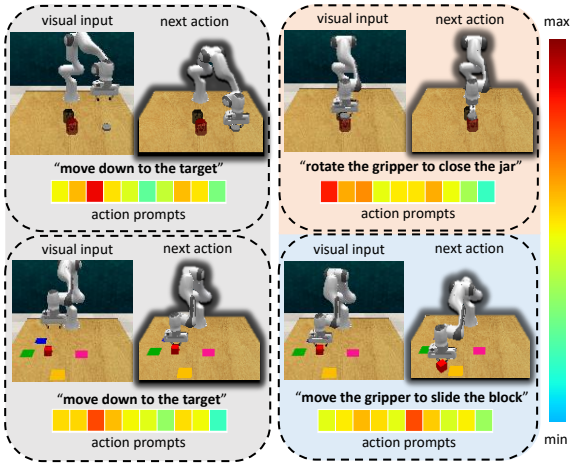

Figure 5. Visualization of skill-specific action prompts by Grad-CAM as the agent executes two manipulation skills: close jar (the first row) and slide block (the second row).

Visualization of the skill-specific action prompts. We visualize our skill-specific action prompts by Grad-CAM [48], as shown in Figure 5. Our experiments train the agent in the B5-5N1 setting with totaling 10 action prompts. We first compute the sum of the parameter gradients for each action prompt and then normalize these values to calculate Grad-CAM weights, displayed in different colors. We show the results across two different skills. When the agent executes the same action, *e.g.* "move down to the target", the third action prompt weight is maximized (see the first column). Furthermore, when performing different actions, different weights of skill-specific action prompts are maximized (see the second column). It demonstrates that action prompts can learn skill-specific action primitives. Freezing old action prompts while learning new ones helps prevent forgetting and adapt to new action primitives.

4.3. Real world experiments

We conduct five manipulation skills in the real-world environment to further validate the effectiveness of our method. We use a Franka Panda robotic arm to execute the action and a Realsense D455 camera to capture RGB-D images as observations. For each training step, we collect 20 demonstrations. The training setup is B1-4N1 where we first train on a base skill, then incrementally add one new skill at a time for training, with a total of four new skills added. During testing, we perform 10 test runs for each learned skill and report the success rate of task execution. **More details about the real world experiments and videos are shown in the supplementary material.**

We compare our method with the baseline [37] without any incremental policy. As shown in Table 5, it is evident that without the incremental strategy, the knowledge of previous skills is rapidly forgotten when training on new skills. After incorporating our incremental strategy, the success rate on new skills is lower than the baseline. This occurs because the baseline has overfitted to the new skill, while our model, which is designed to retain knowledge from previous skills, experiences a slight decrease in its ability to learn new skills. This trade-off is an inherent challenge in incremental learning.

5. Conclusion

In this work, we focus on a new and challenging setting, skill-incremental learning in robotic manipulation, which is to continually learn new skills while maintaining the previously learned skills. We conduct experiments on the RLBench benchmark and find that traditional methods suffer from catastrophic forgetting because they overlook the temporal and action complexities of robotic manipulation. Our approach proposes a temporal replay strategy to address the temporal complexities and an extendable PerceiverIO model with adaptive action prompts to address the action complexities. Extensive experiments demonstrate that our iManip framework excels in effectiveness, robustness, lightweight design, and extendability.

Acknowledgements

This work was supported partially by NSFC (92470202, U21A20471), Guangdong NSF Project (No.2023B1515040025), Guangdong Key Research and Development Program (No.2024B0101040004).

References

[1] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. 2

[2] Arjun Ashok, KJ Joseph, and Vineeth N Balasubramanian. Class-incremental learning with cross-space clustering and controlled transfer. In *Proceedings of the European Conference on Computer Vision*, 2022. 2

[3] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021. 4

[4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 1

[5] 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. *arXiv preprint arXiv:2212.06817*, 2022. 2

[6] Jia-Feng Cai, Zibo Chen, Xiao-Ming Wu, Jian-Jian Jiang, Yi-Lin Wei, and Wei-Shi Zheng. Real-to-sim grasp: Rethinking the gap between simulation and real world in grasp detection. 2024. 1

[7] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision*, 2018. 2, 3, 5, 6, 7

[8] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision*, 2018. 2, 3, 4, 5, 7

[9] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019. 2

[10] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. 1

[11] Shizhe Chen, Ricardo Garcia, Cordelia Schmid, and Ivan Laptev. Polarnet: 3d point clouds for language-guided robotic manipulation. *arXiv preprint arXiv:2309.15596*, 2023. 2

[12] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback. *arXiv preprint arXiv:2407.01512*, 2024. 2

[13] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2023. 2

[14] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Proceedings of the European Conference on Computer Vision*, 2020. 2

[15] Chongkai Gao, Haichuan Gao, Shangqi Guo, Tianren Zhang, and Feng Chen. Cril: Continual robot imitation learning via generative and prediction model. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021. 2

[16] Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation. *arXiv preprint arXiv:2306.17817*, 2023. 2

[17] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, 2023.

[18] Pierre-Louis Guhur, Shizhe Chen, Ricardo Garcia Pinel, Makarand Tapaswi, Ivan Laptev, and Cordelia Schmid. Instruction-driven history-aware policies for robotic manipulations. In *Conference on Robot Learning*, 2023. 2

[19] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 2, 4, 5

[20] Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. Distilling causal effect of data in class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[21] Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024. 1

[22] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, 2021. 2

[23] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 2020. 2, 6

[24] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, 2022. 2

[25] Minsoo Kang, Jaeyoo Park, and Bohyung Han. Class-incremental learning by knowledge distillation with adaptive feature consolidation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[26] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024. 2

[27] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 2023. 2

[28] Khimya Khetarpal, Shagun Sodhani, Sarath Chandar, and Doina Precup. Environments for lifelong reinforcement learning. *arXiv preprint arXiv:1811.10732*, 2018. 2

[29] Byeonghwi Kim, Minhyuk Seo, and Jonghyun Choi. Online continual learning for interactive instruction following agents. *arXiv preprint arXiv:2403.07548*, 2024. 2

[30] 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. 2024. 1

[31] Kibok Lee, Kimin Lee, Jinwoo Shin, and Honglak Lee. Overcoming catastrophic forgetting with unlabeled data in the wild. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2019. 2

[32] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International conference on machine learning*, 2019. 2

[33] Yuan-Ming Li, Ling-An Zeng, Jing-Ke Meng, and Wei-Shi Zheng. Continual action assessment via task-consistent score-discriminative feature distribution modeling. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024. 2

[34] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 2017. 2

[35] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 2024. 1, 2

[36] Yaoyao Liu, Bernt Schiele, and Qianru Sun. Adaptive aggregation networks for class-incremental learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2021. 2

[37] Guanxing Lu, Shiyi Zhang, Ziwei Wang, Changliu Liu, Jiwen Lu, and Yansong Tang. Manigaussian: Dynamic gaussian splatting for multi-task robotic manipulation. In *Proceedings of the European Conference on Computer Vision*, 2024. 2, 3, 5, 6, 7, 8

[38] Xiao Ma, Sumit Patidar, Iain Haughton, and Stephen James. Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 1

[39] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 2022. 2

[40] Jorge Mendez, Shashank Shivkumar, and Eric Eaton. Lifelong inverse reinforcement learning. *Advances in neural information processing systems*, 2018. 2

[41] Yuan Meng, Zhenshan Bing, Xiangtong Yao, Kejia Chen, Kai Huang, Yang Gao, Fuchun Sun, and Alois Knoll. Preserving and combining knowledge in robotic lifelong reinforcement learning. *Nature Machine Intelligence*, 2025. 2

[42] Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 2

[43] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019. 2

[44] Quang Pham, Chenghao Liu, and Steven Hoi. Dualnet: Continual learning, fast and slow. *Advances in Neural Information Processing Systems*, 2021. 2

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 6

[46] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 2, 3, 4, 5, 6, 7

[47] Hyunwoo Ryu, Jiwoo Kim, Hyunseok An, Junwoo Chang, Joohwan Seo, Taehan Kim, Yubin Kim, Chaewon Hwang, Jongeun Choi, and Roberto Horowitz. Diffusion-edfs: Bi-equivariant denoising generative modeling on se (3) for visual robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 1

[48] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, 2017. 8

[49] Lin Shao, Toki Migimatsu, Qiang Zhang, Karen Yang, and Jeannette Bohg. Concept2robot: Learning manipulation concepts from instructions and human demonstrations. *The International Journal of Robotics Research*, 2021. 2

[50] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on robot learning*, 2022. 2

[51] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, 2023. 2, 3, 4, 5, 6

[52] James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always be dreaming: A new approach for data-free class-incremental learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021. 2

[53] Simon Stepputtis, Joseph Campbell, Mariano Phielipp, Stefan Lee, Chitta Baral, and Heni Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. *Advances in Neural Information Processing Systems*, 2020. 2

[54] Yu-Ming Tang, Yi-Xing Peng, and Wei-Shi Zheng. Learning to imagine: Diversify memory for incremental learning using unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[55] Yu-Ming Tang, Yi-Xing Peng, Jingke Meng, and Wei-Shi Zheng. Rethinking few-shot class-incremental learning: Learning from yourself. In *European Conference on Computer Vision*, 2024. 2

[56] Rishabh Tiwari, Krishnateja Killamsetty, Rishabh Iyer, and Pradeep Shenoy. Gcr: Gradient coreset based replay buffer selection for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[57] Yoshihisa Tsurumine and Takamitsu Matsubara. Goal-aware generative adversarial imitation learning from imperfect demonstration for robotic cloth manipulation. *Robotics and Autonomous Systems*, 2022. 1

[58] An-Lan Wang, Nuo Chen, Kun-Yu Lin, Li Yuan-Ming, and Wei-Shi Zheng. Task-oriented 6-dof grasp pose detection in clutters. *arXiv preprint arXiv:2502.16976*, 2025. 2

[59] Yi-Lin Wei, Jian-Jian Jiang, Chengyi Xing, Xian-Tuo Tan, Xiao-Ming Wu, Hao Li, Mark Cutkosky, and Wei-Shi Zheng. Grasp as you say: Language-guided dexterous grasp generation. *arXiv preprint arXiv:2405.19291*, 2024.

[60] Yi-Lin Wei, Mu Lin, Yuhao Lin, Jian-Jian Jiang, Xiao-Ming Wu, Ling-An Zeng, and Wei-Shi Zheng. Afforddexgrasp: Open-set language-guided dexterous grasp with generalizable-instructive affordance. *arXiv preprint arXiv:2503.07360*, 2025. 2

[61] Maciej Wołczyk, Michał Zajac, Razvan Pascanu, Łukasz Kuciński, and Piotr Miłoś. Continual world: A robotic benchmark for continual reinforcement learning. *Advances in Neural Information Processing Systems*, 2021. 2

[62] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024. 2

[63] Xiao-Ming Wu, Jia-Feng Cai, Jian-Jian Jiang, Dian Zheng, Yi-Lin Wei, and Wei-Shi Zheng. An economic framework for 6-dof grasp detection. In *European Conference on Computer Vision*, 2024. 2

[64] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019. 5

[65] Zhou Xian and Nikolaos Gkanatsios. Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. In *Conference on Robot Learning*, 2023. 2

[66] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021. 2

[67] Yanjie Ze, Ge Yan, Yueh-Hua Wu, Annabella Macaluso, Yuying Ge, Jianglong Ye, Nicklas Hansen, Li Erran Li, and Xiaolong Wang. Gnfactor: Multi-task real robot learning with generalizable neural feature fields. In *Conference on Robot Learning*, 2023. 2, 3, 5, 6

[68] Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Co-transport for class-incremental learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, 2021. 2

[69] Jiaming Zhou, Teli Ma, Kun-Yu Lin, Zifan Wang, Ronghe Qiu, and Junwei Liang. Mitigating the human-robot domain discrepancy in visual pre-training for robotic manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025. 1