Never-Ending Embodied Robot Learning

Wenqi Liang [1 2 3] **Gan Sun** [1 2 4] **Qian He** [1 2 3] **Yu Ren** [1 2 3] **Jiahua Dong** [1 2 3] **Yang Cong** [4]

Abstract

Relying on large language models (LLMs), embodied robots could perform complex multi-modal robot manipulation tasks from visual observations with powerful generalization ability. However, most visual behavior-cloning agents suffer from manipulation performance degradation and skill knowledge forgetting when adapting into a series of challenging unseen tasks. We here investigate the above challenge with NBCagent in embodied robots, a pioneering language-conditioned Never-ending Behavior-Cloning agent, which can continually learn observation knowledge of novel robot manipulation skills from skill-specific and skill-shared attributes. Specifically, we establish a skill-specific evolving planner to perform knowledge decoupling, which can continually embed novel skill-specific knowledge in our NBCagent agent from latent and low-rank space. Meanwhile, we propose a skill-shared semantics rendering module and a skill-shared representation distillation module to effectively transfer anti-forgetting skill-shared knowledge, further tackling catastrophic forgetting on old skills from semantics and representation aspects. Finally, we design a continual embodied robot manipulation benchmark, and several expensive experiments demonstrate the significant performance of our method. Visual results, code, and dataset are provided at: https://neragent.github.io.

1. Introduction

Embodied robot learning (ERL) has attracted growing interests in merging machine learning with robot control system to solve various manipulation tasks. With the success of large language models, language-conditioned embodied robot can understand language instructions of users to achieve complex robot tasks and hold significant prospects for applications in industry, healthcare and home robot (Jiang et al., 2022; Brohan et al., 2022). Current language-conditioned behavior-cloning methods focus on address multi-modal data to efficiently execute complex manipulation tasks with visual observations. For instance, PerAct (Shridhar et al., 2023) utilizes a PerceiverIO Transformer (Jaegle et al., 2021) to encode language goals and RGB-D voxel observations, subsequently generating discretized robotic actions. As shown in Fig. 1, they could leverage robust scene understanding capabilities of large language models, and perform task generalization to achieve various novel manipulation tasks.

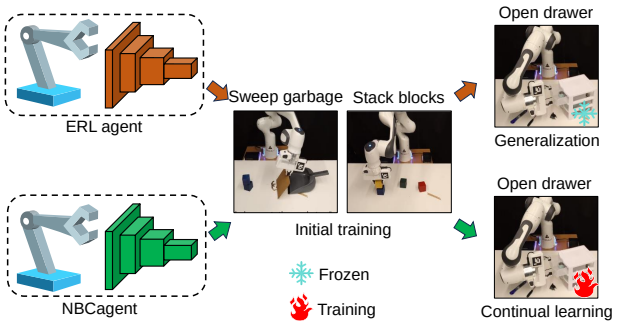

Figure 1. Illustration comparison between ERL and our NBCagent, where ERL performs novel task generalization over a fixed dataset, and our NBCagent can continually learn novel skill knowledge without catastrophic forgetting.

However, the task generalization performance of most existing methods (*e.g.*, PerAct (Shridhar et al., 2023)) is constrained when undertaking never-ending manipulation tasks and handling novel objects with intricate structures. Similar to the way humans acquire skills, this motivates us to enable robots to learn novel challenging manipulation skills in a continual learning manner. For instance, a home robot in the open-ended world is expected to consecutively learn various novel manipulation skills to meet the evolving needs of their owners. A trivial approach for this scenario involves enabling the robot to relearn these complex skills via visual observations, while avoiding forgetting previously acquired skills. However, the majority of existing methods assume training on a fixed dataset and only achieve this via storing old data and retraining model on all data, which leads

[1]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences [2]Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences [3]University of Chinese Academy of Sciences [4]South China University of Technology. Correspondence to: Gan Sun <sungan1412@gmail.com>.

to a large computational burden and high cost of memory storage, thereby being limited in real world.

To address the aforementioned scenarios, we consider a practical challenging embodied robot learning problem, *i.e.*, Never-ending Embodied Robot Learning (NERL), where embodied agent can perform continual learning on successive behavior-cloning manipulation skills and efficiently counteract catastrophic forgetting from old skills. Continual learning (Rebuffi et al., 2017; Yang et al., 2022; Sun et al., 2023) aims to continuously acquire knowledge from a successive data stream and has achieved remarkable performance in several computer vision tasks, such as image classification, object detection, image generation and so on. A naive solution for the NERL problem is to directly integrate continual learning and embodied robot learning together. However, different from learning category-wise knowledge focused by most existing methods, some attributes about skill-wise knowledge should be considered:

• **Skill-Specific Attribute** originates from distinct manipulation sequences, object recognition, and scene understanding inherent in each unique skill. However, current continual learning methods neglect this attribute, which constrains their ability to continually learning novel skills.

• **Skill-Shared Attribute** indicates that different skills possess shared knowledge. For instance, similar object recognition and scene understanding exist between different skills such as stacking bottles and opening bottle caps. Transferring skill-shared knowledge plays a key role in addressing skill-wise knowledge forgetting.

To tackle the above-mentioned challenges, we propose a pioneering language-conditioned Never-ending Behavior-Cloning agent (*i.e.,* NBCagent), which continually acquire skill-wise knowledge from skill-specific and skill-shared attributes with visual observations. To the best of our knowledge, this is an earlier attempt to explore never-ending embodied robot learning for multi-modal behavior-cloning robotic manipulation. To be specific, we propose a skill-specific evolving planner (SEP) to decouple the skill-wise knowledge in the latent and low-rank space, and focus on skill-specific knowledge learning. Additionally, we design a skill-shared semantics rendering module (SSR) and a skill-shared representation distillation module (SRD) to transfer skill-shared knowledge from semantics and representation aspects, respectively. Supervised by Neural Radiance Fields (NeRFs) and a vision foundation model, the SSR can complete skill-shared semantics in 3D voxel space across novel and old skills. Furthermore, the SRD can effectively distill knowledge between old and current models to align skill-shared representation. Several major contributions of our work are as follows:

- We take the earlier attempt to explore a novel real-world challenging problem called Never-ending Embodied Robot Learning (NERL), where we propose Never-ending Behavior-Cloning agent (*i.e.,* NBCagent) to address the core challenges of skill-wise knowledge learning from skill-specific and skill-shared attributes.
- We design a skill-specific evolving planner to decouple the skill-wise knowledge and continually embed skill-specific novel knowledge in our NBCagent. Moreover, a skill-shared semantic rendering module and a skill-shared representation distillation module is developed to learn skill-shared knowledge from semantics and representation aspects, respectively, and further overcome catastrophic forgetting.
- We present a continual embodied robot manipulation benchmark for home robotic manipulation, which consists of two manipulation scenes, kitchen and living room. Qualitative experiments demonstrate the effectiveness and robustness of our proposed NBCagent.

2. Related Work

Robotic Manipulation. Recent works (Brohan et al., 2023; Chowdhery et al., 2023; Huang et al., 2022; Shah et al., 2023; Driess et al., 2023; Brohan et al., 2022; Zitkovich et al., 2023) have resulted in substantial advancements in the accomplishment of intricate tasks. VIMA (Jiang et al., 2022) proposes a novel multi-modal prompting scheme, which transforms a wide range of robotic manipulation tasks into a sequence modeling problem. Compared with directly using images as the manipulation input (Goyal et al., 2022; Shridhar et al., 2022), voxelizing 3D point clouds as a 3D representation (Shridhar et al., 2023; Ze et al., 2023; Goyal et al., 2023; James et al., 2022) can accomplish more complex tasks. PerAct (Shridhar et al., 2023) enables agent to perform better in robotic manipulation by voxelizing RGB-D images and discretizing output actions.

Continual Learning. Continual learning provides the foundation for the adaptive development of AI systems (Wang et al., 2023). The main approaches of continual learning can be categorized into three directions: **Parameter regularization-based** methods (Rebuffi et al., 2017; Li & Hoiem, 2017; Derakhshani et al., 2021; Douillard et al., 2020; Dong et al., 2023) balance the old and new tasks by adding more explicit regularization terms. **Architecture-based** methods (Jung et al., 2020; Wu et al., 2021; Wang et al., 2022; Toldo & Ozay, 2022) construct network parameters for different tasks. **Replay-based** methods include empirical replays (Bang et al., 2021; Rebuffi et al., 2017; Sun et al., 2022; Tiwari et al., 2022) and generative replays (Li et al., 2022; Xiang et al., 2019).

Some works focus on the improvement of robots by continual learning (Ayub & Wagner, 2023; Gao et al., 2021; Hafez & Wermter, 2023; Ayub & Fendley, 2022; Ayub

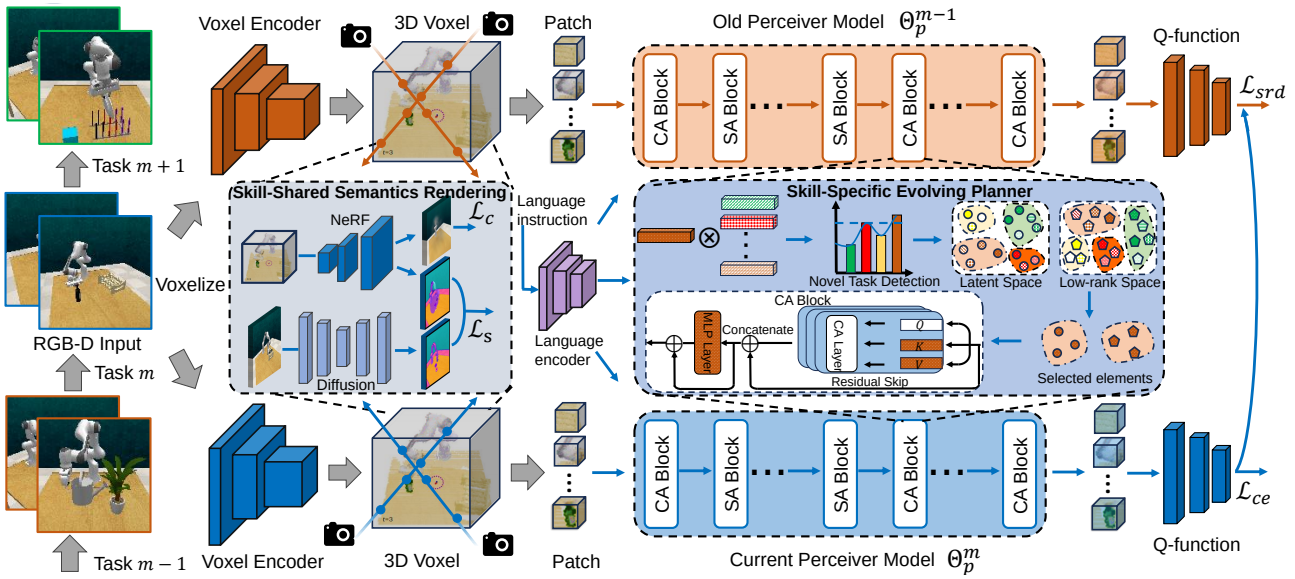

Figure 2. Overview of the proposed NBCagent, where the perceiver model can continually learn novel manipulation skills. A *skill-specific evolving planner* is designed to learn skill-specific knowledge from latent and low-rank space. Meanwhile, we develop a *skill-shared semantics rendering module* and *skill-shared representation distillation loss* $\mathcal{L}_{srd}$ to transfer skill-shared knowledge from semantics and representation aspects.

et al., 2023). LOTUS (Wan et al., 2023) stores a few human demos of novel tasks into the growing skill library, enabling lifelong learning ability for robots. However, these methods cannot be applied to perform language-conditioned behaviour-cloning manipulation as they are incapable of handling multi-modal skill data. Furthermore, they neglect two inherent attributes in learning skill-wise knowledge, and suffer catastrophic forgetting on old skills.

3. Methodology

3.1. Problem Definition and Overview

Problem Definition. Following traditional continual learning methods (Rebuffi et al., 2017; Douillard et al., 2020), we define a multi-modal skill data stream as $\mathcal{T} = \{\mathcal{T}^m\}_{m=1}^{M}$, where M denotes the number of incremental tasks. Each incremental task consists of various robotic manipulation skills, resulting in a total of N^m manipulation skills. The m-th incremental task $\mathcal{T}^m = \{\mathcal{D}_i^m\}_{i=1}^{N^d}$ consists of N^d skill demonstrations. Specifically, each skill demonstration can be extracted to a set of keyframe actions, *i.e.*, $\mathcal{D}_i^m = \{\mathbf{k}_j^{m,i}\}_{j=1}^{N^k}, \mathbf{k}_j^{m,i} = \{\mathbf{a}_j^{m,i}, \mathbf{r}_j^{m,i}, \mathbf{l}^{m,i}\}$, where N^k is the total keyframe action quantity. Given the current state in action space $\mathbf{a}$, the structured observation $\mathbf{r}$ and language instruction $\mathbf{l}$, the agent is expected to predict the next best keyframe action, which can be served as an action classification task (James et al., 2022). Additionally, the structured observation $\mathbf{r}$ is composed of the RGB-D images captured by a single front camera. An action state $\mathbf{a}$ can be divided into a discretized translation $\mathbf{a}_{tran} \in \mathbb{R}^3$, rotation $\mathbf{a}_{rot} \in \mathbb{R}^{(360/5)\times 3}$, gripper open state $\mathbf{a}_{grip} \in \{0, 1\}$, and collision avoidance $\mathbf{a}_{col} \in \{0, 1\}$. Here the rotation parameter $\mathbf{a}_{rot}$ entails the discretization of each rotation axis into a set of R = 5 bins. The collision avoidance parameter $\mathbf{a}_{col}$ provides guidance to the agent regarding the imperative need to avoid collisions.

Overview. The overview of our NBCagent to learn skill-wise knowledge is shown in Fig. 2. When observing a novel incremental task $\mathcal{T}^m$, we initialize the perceiver model Θ_p^m for the current task utilizing the model Θ_p^{m-1} obtained from the last task and store Θ_p^{m-1} as a teacher model to perform our SRD module (we refer to PerceiverIO (Jaegle et al., 2021) as perceiver model for brevity). Following ER (Chaudhry et al., 2019), we build a memory buff $\mathcal{M}$ to replay only few samples form the previous tasks. In the m-th task, our NBCagent aims to execute all learned multi-modal robotic manipulation skills, achieved through iteratively optimizing the model on $\mathcal{T}^m$ and $\mathcal{M}$. Specifically, as shown in Fig. 2, given a RGB-D and language input, Θ_p^m first encode RGB-D input to obtain a deep 3D voxel utilizing a scaled-down voxel encoder $\mathcal{E}_v^m$. Then, we design a SSR module to transfer and complete skill-shared semantics across novel and old skills, which can effectively address catastrophic forgetting on old skills. After that, the patched voxel and language embeddings are sent to cross-attention blocks and self-attention blocks to perform feature extraction and semantics fusion, where we propose SEP to learn skill-specific knowledge from latent and low-rank space. Finally, we utilize a Q-function head to predict the state of the next keyframe in voxel space, where we develop a SRD loss $\mathcal{L}_{srd}$ to tackle catastrophic forgetting by aligning skill-shared representation.

3.2. Skill-Specific Evolving Planner

Aiming at learning category-wise knowledge, existing continual learning methods (Rebuffi et al., 2017; Douillard et al., 2020) assume that the knowledge acquired from novel tasks and that from previous tasks are mutually independent. Differently, for learning skill-wise knowledge, we consider decoupling the knowledge to the skill-shared knowledge and skill-specific knowledge. For instance, an embodied agent aims to stack the wine bottle after learning to open the wine bottle. It does not require relearning the skill-shared knowledge of scene understanding and object recognition; instead, the agent is expected to focus on acquiring the skill-specific knowledge related to the operation sequence of stacking and reducing forgetting on skill-shared knowledge. Considering this motivation, we design a skill-specific evolving planner (SEP) to perform knowledge decoupling, enabling effectively continual learning of novel skills. Specifically, we first develop an adaptive language semantic bank to retrieve the skill-specific language semantic information. In light of this information, SEP can encode the multi-modal input from skill-wise latent and low-rank space, and learn the novel knowledge through a skill-specific network.

First of all, when observing a novel skill, we utilize a language encoder $\mathcal{E}_c$ from CLIP (Radford et al., 2021) to encode the language instruction. This can obtain a skill-specific language semantic information $\mathbf{l}_s \in \mathbb{R}^{D^c}$, and text token information $\mathbf{l}_x \in \mathbb{R}^{N^c \times D^x}$, *i.e.*, $\mathbf{l}_s, \mathbf{l}_x = \mathcal{E}_c(\mathbf{l})$, where D^s and D^x represent the dimension of $\mathbf{l}_s$ and $\mathbf{l}_x$, and N^c denotes the token length. Then, we compensate semantic information for our adaptive language semantic bank $\mathcal{B}$ during training via an exponential moving average strategy:

$$\mathcal{B}[I,:] = (1 - \mathcal{C}_{max})\mathcal{B}[I,:] + \mathcal{C}_{max}\mathbf{l}_s, \tag{1}$$

where $\mathcal{B} \in \mathbb{R}^{N^b \times D^c}$ is initialized by N^b zero vectors. $\mathcal{C} \in \mathbb{R}^{N^b}$ is the cosine similarity matrix between the sentence information $\mathbf{l}_s$ and each vector in $\mathcal{B}$. if $\mathcal{C}_{max} > \delta$, we set $I = \text{argmax}(\mathcal{C})$; otherwise, we set $I = \text{nonzero}(\mathcal{B}) + 1$ and $\mathcal{C}_{max} = 1$, where δ is a fixed threshold, and $\text{nonzero}(\cdot)$ is a operation employed to calculate the number of nonzero vectors in $\mathcal{B}$. To this end, we can obtain the skill-wise code I for the mini-batch training data, which plays a key role in performing skill-specific network training in the following.

Furthermore, different from existing multi-modal ERL methods (Shridhar et al., 2023; Ze et al., 2023) that only encode multi-modal input from a skill-shared latent space, our NBCagent considers to develop a dynamic skill-specific latent space $\mathbf{S} \in \mathbb{R}^{N^s \times N^l \times D^s}$, where N^s denotes the number of learned skills, and N^l represents the learnable latent vector quantity. NBCagent encodes these latent vectors with the multi-modal input utilizing a cross-attention layer to obtain the final feature. Specifically, following (Shridhar et al., 2023), we first to apply a scaled-down 3D convolution encoder to patch and encode a voxel observation $\mathbf{v}$ to obtain $\hat{\mathbf{v}}$, where $\mathbf{v}$ is obtained by voxelization process from $\mathbf{r}$. Then, we concatenate the encoded proprioception of agent in current state $\mathbf{a}$ and voxel observation $\hat{\mathbf{v}}$ to obtain $\mathbf{p} \in \mathbb{R}^{N^p \times D}$. In light of this, we employ a cross-attention layer to perform semantics interaction and obtain the cross-attention feature $\mathbf{F_c} \in \mathbb{R}^{(N^p+N^c)\times D}$:

$$\mathbf{F}_c = \rho(\frac{\text{Cat}(\mathbf{p}, \mathbf{c}_x)\mathbf{W}_q(\mathbf{S}[I,:]\mathbf{W}_k)^\top}{\sqrt{d}})(\mathbf{S}[I,:]\mathbf{W}_v), \tag{2}$$

where $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{D\times D}$ are linear projection layers, and $\text{Cat}(\cdot)$ denotes the concatenation operation. ρ indices the softmax function and d is a scaling factor. Then, we apply two additional linear projection layers $\mathbf{W}_o$ to handle the cross-attention feature $\mathbf{F}_c$. Similarly, we apply a series of self-attention blocks and a cross-attention decoder to further extract feature. In light of this, NBCagent can continually encode the novel multi-modal input from a skill-specific latent space, which is beneficial to learn some skill-specific knowledge from latent space.

Considering the limitation in representing skill-specific knowledge from latent space, we further explore to learn skill-specific knowledge from low-rank space. Specifically, we introduce a low-rank adaptation layer (LoRA) (Hu et al., 2021) that can learn skill-specific knowledge in an efficient manner. For $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_o$ in each attention block, we design a set of skill-specific LoRA layers to perform skill-specific forward and obtain the final output feature $\mathbf{F}_x$ as follows:

$$\mathbf{F}_x = \mathbf{X}\mathbf{W} + \mathbf{X}\mathbf{W}_r[I,:] = \mathbf{X}\mathbf{W} + \mathbf{X}\mathbf{W}_a[I,:]\mathbf{W}_b[I,:], \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{D\times D}$ represents one of $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_o$ and $\mathbf{X} \in \mathbb{R}^{(N^v+N^p)\times D}$ denotes the input feature for these

Algorithm 1 Pipeline of Our ESP.

Require: Initialized adaptive language semantic bank $\mathcal{B}$ with N^b zero vectors; Initialized dynamic skill-specific latent space $\mathbf{S} = \emptyset$; Initialized low-rank space $\mathbf{W} = \emptyset$; The hyper-parameter δ;
Input: language embedding $\mathbf{l}_s$;
Output: $\mathbf{S}[I,:], \mathbf{W}[I,:]$;

```
Compute cosine similarity matrix C between B and l_s;
if C_max > δ then
    I ← argmax(C);
    Update B by Eq. (1);
    Return: S[I,:], W[I,:];
else
    I ← nonzero(B + 1);
    Expand B by Eq. (1);
    Randomly initialize S[I,:], W[I,:];
    Return: S[I,:], W[I,:];
end if
```

Algorithm 2 Optimization Pipeline of Our NBCagent.

Require: Robotics incremental tasks $\{\mathcal{T}^m\}_{m=1}^{M}$ with datasets $\mathcal{T}^m = \{\mathcal{D}^m\}_{i=1}^{N^d}$; Initialized perceiver model: Θ_p^0; Initialized NeRF model: Θ_n^0; Pre-trained diffusion model: Θ_u; Pre-trained CLIP language encoder: $\mathcal{E}_c$; Initialized memory buff: $\mathcal{M} = \emptyset$; Iterations: $\{\mathcal{I}^m\}_{m=1}^{M}$.

#While observing a new task $\mathcal{T}^m$:
for $z = 1, 2, \cdots, \mathcal{I}^m$ **do**
  Randomly select keyframe $\mathbf{k}_j^m$ from $\{\mathcal{D}_i^m\}_{i=1}^{N^d} \cup \mathcal{M}$;
  Obtain $\mathrm{v}^m, \mathrm{v}^{m-1}, \mathrm{l}_s, \mathrm{l}_x$ utilizing $\mathcal{E}_v^m$, $\mathcal{E}_v^{m-1}$and $\mathcal{E}_c$;
  Compute $\mathcal{L}_{ssr}$ by SSR $(\mathrm{v}^m, \mathrm{v}^{m-1}, \mathrm{l}_x, \Theta_u)$ using Eq. (4) and Eq. (12);
  $\mathbf{S}[I,:], \mathbf{W}[I,:] \leftarrow$ ESP $(\mathbf{l}_s)$;
  Compute $\mathcal{L}_{ce}, \mathcal{L}_{srd}$ utilizing $\mathbf{S}[I,:]$, $\mathbf{W}[I,:]$, $\Theta_p^m, \Theta_p^{m-1}$;
  Update Θ_p^m by Eq. (15);
end for
Store few samples from $\{\mathcal{D}_i^m\}_{i=1}^{N^d}$ in $\mathcal{M}$;
Return: $\Theta_p^m, \mathcal{M}$.

projection layers. $\mathbf{W}_r[I,:] = \mathbf{W}_a[I,:]\mathbf{W}_b[I,:]$ is a low-rank decomposition, where $\mathbf{W}_a \in \mathbb{R}^{N^l \times D \times N^r}$ is initialized in a random Gaussian manner, $\mathbf{W}_b \in \mathbb{R}^{N^l \times N^r \times D}$ is initialized by zero and N^r is a hyper-parameter controlling the size of LoRA layers. On the one hand, obviously, $\mathbf{W}$ is shared among all skills and expected to learn skill-shared knowledge. On the other hand, each $\mathbf{W}_r$ is executed to learn different novel skills and perform skill-specific forward, resulting in continually embedding skill-specific knowledge to our NBCagent. Specifically, we summary the process of our SEP in **Algorithm** 1.

3.3. Skill-Shared Semantics Rendering Module

For language-conditional behaviour-cloning manipulation, a comprehensive semantics understanding of the 3D scene (Driess et al., 2022) plays a key role in enabling agent to perform complicated manipulation skills. Especially in NERL problem, there exist skill-shared semantics across various skills, such as 3D object and scene semantics. The existence of forgetting on semantic space makes these semantics incomplete, further resulting in catastrophic forgetting on old skills. Considering this motivation, we develop a skill-shared semantics rendering module (SSR) to transfer skill-shared semantic information of 3D voxel space, where a NeRF model and vision foundation model provide semantics supervision to effectively enrich the 3D voxel semantics. Drawing inspiration from 3D visual representation learning methods (Shim et al., 2023; Ze et al., 2023), we leverage a latent-conditioned NeRF architecture (Yu et al., 2021) not only to synthesizes RGB color $\mathbf{c}$ of a novel image views like traditional NeRF (Mildenhall et al., 2021), but also to render the semantic feature $\mathbf{s}$ from 3D voxel space as follows:

$$\mathcal{F}_{\Theta_n^m}(\mathbf{x}, \mathbf{d}, \mathbf{v}_s) = (\sigma, \mathbf{c}, \mathbf{s}), \tag{4}$$

where $\mathcal{F}_{\Theta_n^m}$ denotes the neural rendering function of NeRF model Θ_n^m. The 3D voxel feature $\mathbf{v}_s$ is obtained by a grid sample method based on trilinear interpolation from the 3D voxel observation $\mathbf{v}$. $\mathbf{x}$, σ is the 3D input point and differential density, and $\mathbf{d}$ represents unit viewing direction. The camera ray $\mathbf{r}$ can be obtained by: $\mathbf{r} = \mathbf{o} + t\mathbf{d}$, where $\mathbf{o}$ indicates the camera origin. By adding field-wise branches, Θ_n^m performs the same neural rendering function $\mathcal{F}_{\Theta_n^m}$ to estimate RGB color $\mathbf{c}$ and semantic feature $\mathbf{s}$. Thus, the same accumulated transmittance $T(t)$ is shared to predict the two different fields and is defined as follows:

$$T(t) = \exp(-\int_{t_n}^{t} \sigma(s)ds). \tag{5}$$

In light of this, a RGB image $\mathbf{C}$ and 2D semantic map $\mathbf{S}$ can be rendered as :

$$\mathbf{C}(\mathbf{r}, \mathbf{v}_s) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}, \mathbf{v}_s)\mathbf{c}(\mathbf{r}, \mathbf{d}, \mathbf{v}_s)dt, \tag{6}$$

$$\mathbf{S}(\mathbf{r}, \mathbf{v}_s) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}, \mathbf{v}_s)\mathbf{s}(\mathbf{r}, \mathbf{d}, \mathbf{v}_s)dt. \tag{7}$$

To distill skill-shared knowledge, we initialize a NeRF model acquired from the last task and denote it as Θ_n^{m-1}. Then, we feed the same input to obtain the pseudo ground truth $\hat{\mathbf{C}}$ by Eq. 6. We design a loss function to supervise the reconstruction process as follows:

$$\mathcal{L}_c = \sum_{\mathbf{r}\in\mathcal{R}} \|\mathbf{C}(\mathbf{r}, \mathbf{v}_s) - \mathbf{Y}_c(\mathbf{r})\|_2^2 + \beta \sum_{\mathbf{r}\in\mathcal{R}} \|\mathbf{C}(\mathbf{r}, \mathbf{v}_s) - \hat{\mathbf{C}}(\mathbf{r}, \mathbf{v}_s)\|_2^2 \cdot \mathbb{I}_{\mathbf{v}_s \notin \mathcal{T}^m}, \tag{8}$$

where $\mathbf{Y}_c$ indicates the ground truth color and $\mathcal{R}$ is the set of all camera rays. α is the hyper-parameter to control the weight of loss function. $\mathbb{I}_{\mathbf{v}_s \notin \mathcal{T}^m}$ is defined such that $\mathbb{I}_{\mathbf{v}_s \notin \mathcal{T}^m} = 1$ when the condition $\mathbf{v}_s \notin \mathcal{T}^m$ is satisfied, and $\mathbb{I}_{\mathbf{v}_s \notin \mathcal{T}^m} = 0$ otherwise.

Considering the insufficiency in capturing skill-shared semantics by reconstructing novel views, we introduce a pre-trained visual foundation model that contains robust scene semantics to provide supervision. Relying on being pre-trained on large-scale vision-language dataset, Stable Diffusion model (Rombach et al., 2022) can possess robust intrinsic representational capabilities and are consequently utilized for semantic representation in segmentation and classification tasks (Xu et al., 2023; Li et al., 2023). In light of this, we employ a text-to-image Stable Diffusion model Θ_u to extract vision-language semantic feature for supervising. Specifically, given a input view, *i.e.*, $\mathbf{Y}_c$, we perform

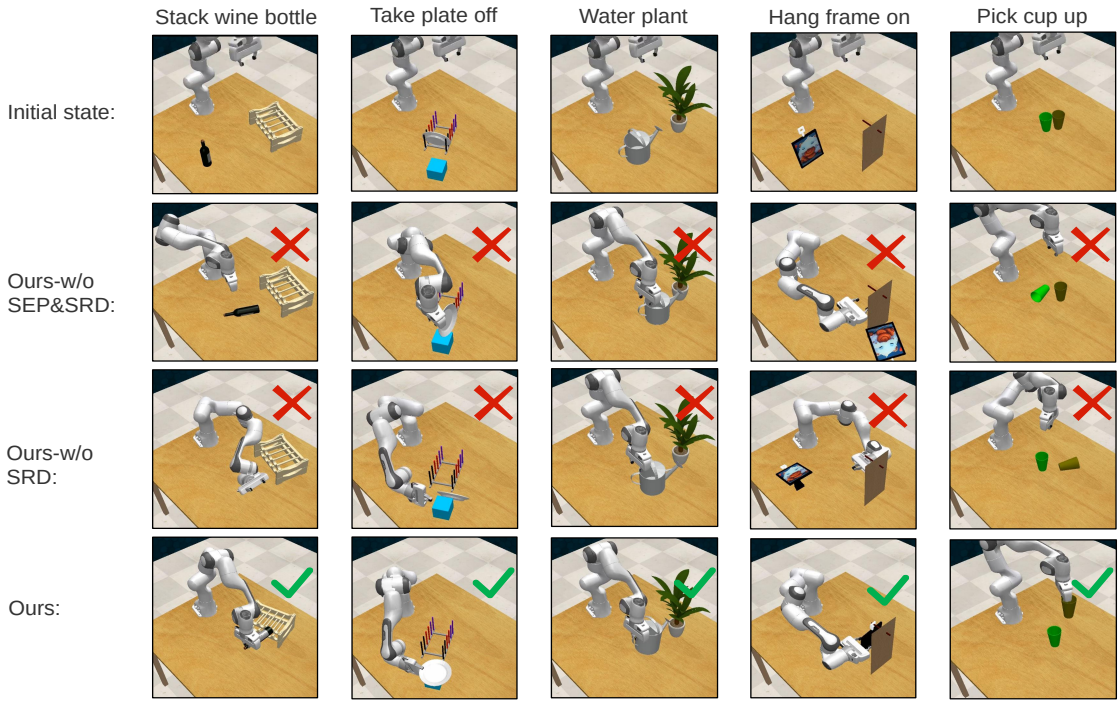

Figure 3. Visualization of prediction results on various manipulation skills between Ours, Ours-w/oSRD and Ours-w/oSEP&SRD.

a one-step noise adding process to obtain a a noisy image $\mathbf{Y}_{c,t}$. Then we utilize our diffusion model Θ_u to collect vision-language semantic feature as the ground truth $\hat{\mathbf{S}}$:

$$\mathbf{Y}_{c,t}(\mathbf{r}) := \sqrt{\alpha_t}\mathcal{E}_v(\mathbf{Y}_c(\mathbf{r})) + \sqrt{1-\alpha_t}\epsilon, \tag{9}$$

$$\hat{\mathbf{S}}(\mathbf{r}, \mathbf{l}_p) = \Theta_u(\mathbf{Y}_{c,t}(\mathbf{r}), \mathcal{E}_c(\mathbf{l}_p)), \tag{10}$$

where $\mathcal{E}_v$ is a VAE encoder to encode image $\mathbf{Y}_c$ from pixel space to latent semantic space. t represents the diffusion process step, $\epsilon \sim \mathcal{N}(0,1)$ and α_t is designed to control the noise schedule. $\mathbf{l}_p$ denotes the language prompt modified from task description $\mathbf{l}$. To this end, we align the rendered semantic feature $\mathbf{S}$ and diffusion feature $\hat{\mathbf{S}}$ to perform semantics transfer as follows:

$$\mathcal{L}_s = \sum_{\mathbf{r}\in\mathcal{R}} \|\mathbf{S}(\mathbf{r}, \mathbf{v}_s) - \hat{\mathbf{S}}(\mathbf{r}, \mathbf{l}_p)\|_2^2. \tag{11}$$

In summary, the major objective of our SSR module to complete skill-shared semantics can be expressed as:

$$\mathcal{L}_{srr} = \mathcal{L}_c + \lambda_1 \mathcal{L}_s, \tag{12}$$

where λ_1 is the hyper-parameter.

3.4. Skill-Shared Representation Distillation Module

To address catastrophic forgetting on old skills, we develop a skill-shared representation distillation module (SRD) to align skill-shared representation, as presented in Fig. 2. Specifically, given a multi-modal keyframe input $\mathbf{k}_j^m = \{\mathbf{a}_j^m, \mathbf{r}_j^m, \mathbf{l}^m\}$ from a mini batch, we can obtain a keyframe prediction $\mathbf{P}^m(\mathbf{k}_j^m, \Theta_p^m) = \{\mathbf{P}_{j,tran}^m, \mathbf{P}_{j,rot}^m, \mathbf{P}_{j,grip}^m, \mathbf{P}_{j,col}^m\}$. To supervise our NER-agent to learn skill-wise knowledge, we follow (Shridhar et al., 2023) to introduce a cross-entropy loss for optimizing as follows:

$$\mathcal{L}_{ce} = -\frac{1}{B}\sum_{j=1}^{B} \mathbf{Y}_j^m \log(\mathbf{P}^m(\mathbf{k}_j^m, \Theta_p^m)), \tag{13}$$

where B represents the batch size, and $\mathbf{Y}_j^m = \{\mathbf{Y}_{j,tran}^m, \mathbf{Y}_{j,rot}^m, \mathbf{Y}_{j,grip}^m, \mathbf{Y}_{j,col}^m\}$ is the ground truth for predicting the next keyframe. In light of this, NBCagent can continually learn skill-wise knowledge from current dataset $\mathcal{T}^m$ and memory buff $\mathcal{M}$. However, due to the limited amount of data available from old skills in memory buff $\mathcal{M}$, the data imbalance occurs between novel and old skills, further resulting in overfitting to learn novel skill-wise knowledge and forgetting skill-shared knowledge on old skills.

To address the aforementioned problems, we take an attempt to employ knowledge distillation in NERL problem. Specifically, we initialize a teacher model with the perceiver model Θ_p^{m-1} from the last task to extract the soft label: $\hat{\mathbf{Y}}_j^m = \mathbf{P}^m(\mathbf{k}_j^m, \Theta_p^{m-1}), \hat{\mathbf{Y}}_j^m = \{\hat{\mathbf{Y}}_{j,tran}^m, \hat{\mathbf{Y}}_{j,rot}^m, \hat{\mathbf{Y}}_{j,grip}^m, \hat{\mathbf{Y}}_{j,col}^m\}$ and apply the Kullback-Leibler divergence to align the outputs of two agents as follows:

$$\mathcal{L}_{srd} = \frac{1}{\hat{B}}\sum_{j=1}^{B} \rho(\hat{\mathbf{Y}}_j^m/\tau)\log\left(\frac{\rho(\hat{\mathbf{Y}}_j^m/\tau)}{\rho(\mathbf{P}^m(\mathbf{k}_j^m, \Theta_p^m)/\tau)}\right) \cdot \mathbb{I}_{\mathbf{k}_j^m \notin \mathcal{T}^m}, \tag{14}$$

Table 1. Comparisons of success rate (%) on Kitchen and Living Room. **Red** and **Blue** represents the highest results and runner-up.

Comparison Methods	**5-5** (2 steps)					**5-1** (6 steps)					**6-3** (3 steps)					**6-2** (4 steps)				
	1-5	6-10	All	Avg.	For.	1-5	6-10	All	Avg.	For.	1-6	7-12	All	Avg.	For.	1-6	7-12	All	Avg.	For.
PerAct	58.9	30.1	44.5	–	–	58.9	30.1	44.5	–	–	34.7	27.3	31.0	–	–	34.7	27.3	31.0	–	–
GNFactor	56.3	32.3	44.3	–	–	56.3	32.3	44.3	–	–	48.0	32.0	40.0	–	–	48.0	32.0	40.0	–	–
Fine-Tuning	15.7	26.4	21.1	38.9	41.1	3.2	20.0	9.6	13.8	39.1	4.4	29.8	17.1	29.0	44.1	6.2	19.1	12.7	24.2	43.9
ER	**56.0**	25.6	40.8	**50.0**	**3.2**	53.6	29.6	41.6	49.7	9.8	**43.8**	31.1	37.4	42.2	**7.7**	**40.7**	30.2	35.4	38.1	17.6
Ours-w/oSEP&SRD	**56.0**	26.7	**41.3**	49.1	**0.8**	**56.8**	30.4	43.6	50.2	**7.1**	42.0	33.1	37.6	45.3	**12.6**	38.4	35.6	37.0	40.9	**16.5**
Ours-w/oSRD	41.1	**38.1**	39.6	49.8	18.9	48.0	**41.6**	**44.8**	**53.9**	14.7	41.1	**34.4**	**37.8**	**45.9**	15.6	40.4	**39.1**	**39.8**	**43.3**	19.1
Ours	**53.6**	**36.3**	**44.9**	**52.5**	6.4	**54.4**	**37.6**	**46.0**	**55.2**	**8.9**	**44.9**	**42.2**	**43.6**	**47.6**	**7.7**	**45.8**	**35.3**	**40.6**	**43.5**	**10.9**

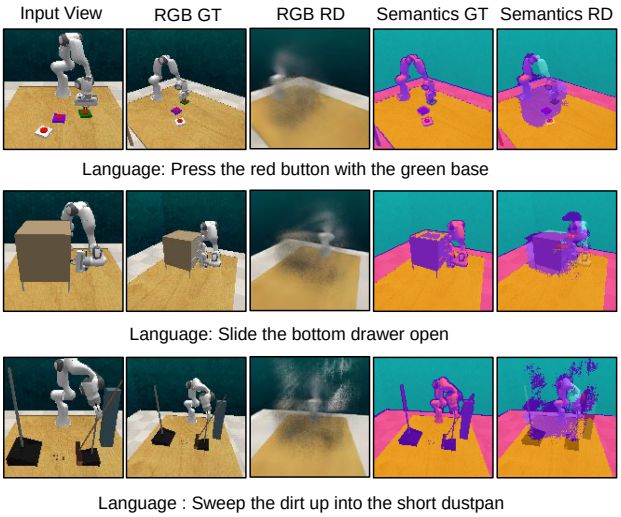

Figure 4. Visualization of rendering results in our SSR module. RGB GT denotes the color ground truth, and Semantics GT represents the semantics ground truth extracted by Stable Diffusion model. RGB RD and Semantics RD are the rendering novel view and semantic feature.

where $\hat{B} = \sum_{j=1}^{B} \mathbb{I}_{\mathbf{k}_j^m \notin \mathcal{T}^m}$, and τ is a temperature hyper-parameter.

In conclusion, to perform skill-specific knowledge learning, we first develop SEP to accumulate skill-specific knowledge on latent and low-rank space, thereby effectively learning novel skills. Furthermore, SSR and SRD modules are designed to transfer skill-shared knowledge from semantics and representation aspects, resulting in efficiently tackling old skill forgetting. The optimization of our NBCagent can be simplified as:

$$\mathcal{L}_{total} = \mathcal{L}_{ce} + \mathcal{L}_{srr} + \lambda_2 \mathcal{L}_{srd}. \tag{15}$$

4. Experiments

4.1. Implementation Details

Dataset. Following PerAct, we conduct our experiments on RLBench (James et al., 2020) and simulate in CoppelaSim (Rohmer et al., 2013). To simulate the working scenarios of NERL, we design two NERL benchmark datasets, called **Kitchen** and **Living Room**. Specifically, **Kitchen** is constructed by gathering 10 manipulation skills pertinent to kitchen environments, and **Living Room** consists of 12 manipulation skills associated with living room scenarios. Each manipulation skill includes a training set of 20 episodes and a test set of 25 episodes. Furthermore, these skills involve various variations encompassing randomly sampled attributes such as colors, sizes, counts, placements, and object categories, resulting in a total of 101 distinct variations. We provide comprehensive details of two benchmark datasets in **Appendix A**.

Baselines. We conduct the comprehensive evaluation between our NBCagent and the following four methods: PerAct (Shridhar et al., 2023), GNFactor (Ze et al., 2023), Fine-Tuning and ER (Chaudhry et al., 2019). PerAct and GNFactor joint train all manipulation skills within one-stage training, for two datasets respectively, referred to as the upperbound. Fine-Tuning achieves continual learning by fine-tuning all parameters in perceiver model on novel skills. ER randomly stores old skill data to memory buff and replay them when detecting novel skills.

Training Details. In NERL, we assume that the agent undergoes initial learning through a set of manipulation skills, referred to as base task, while characterizing novel skills as incremental tasks. On **Kitchen** dataset, the base task includes 5 manipulation skills, and each incremental task consists of 1 manipulation skill (total 6 steps) and 5 manipulation skills (total 2 steps), marked as **5-1** and **5-5**. Likewise, on **Living Room** dataset, 12 manipulation skills are divided into two NERL settings: **6-3** (total 3 steps) and **6-2** (total 4 steps). In addition, the LAMB (You et al., 2019) optimizer is applied for all methods with a initial learning rate of 5.0×10^{-4} and a batch size of 2. We utilize 100K training iterations for PerAct and GNFactor, 80K for base task training, 20K for incremental task training. We store a fixed 4 episodes of each old skill in $\mathcal{M}$ for ER and Ours. Additional training details are available in **Appendix B**.

Evaluation Metric. Following ERL methods (Ze et al., 2023; Goyal et al., 2023), we use the success score (%) $\mathcal{I}_i^m$ as the basic indicator for evaluation, where $\mathcal{I}_i^m$ represents the success score of i-th manipulation skill at m-th incremental task. Specifically, we first compute the mean

Table 2. Comparison results on Kitchen under the setting of **5-1**. **Red** and **Blue** represents the highest results and runner-up.

Comparison Methods	1	2	3	4	5	6	7	8	9	10	All	Imp.
PerAct	96.0	77.3	53.3	30.7	37.3	52.0	2.7	18.7	4.0	73.3	44.5	⇑ 1.5
GNFactor	92.0	60.0	70.7	12.0	46.7	52.0	5.3	22.7	10.7	70.7	44.3	⇑ 1.7
Fine-Tuning	17.3	0.0	0.0	0.0	0.0	8.0	0.0	0.0	9.3	**64.0**	9.7	⇑ 36.3
ER	**90.7**	54.7	37.3	**54.7**	28.0	60.0	5.3	0.0	**24.0**	**62.7**	41.6	⇑ 4.4
Ours-w/oSEP&SRD	86.7	49.3	34.7	**65.3**	**48.0**	**66.7**	9.3	**24.0**	5.3	46.7	43.6	⇑ 2.4
Ours-w/oSRD	68.0	**65.3**	**50.7**	25.3	33.3	**76.0**	**18.7**	**24.0**	**22.7**	**64.0**	**44.8**	⇑ 1.2
Ours	**92.0**	**61.3**	**44.0**	34.7	**41.3**	**76.0**	**17.3**	**26.7**	14.7	52.0	**46.0**	–

Table 3. Comparison results on Living Room under the setting of **6-3**. **Red** and **Blue** represents the highest results and runner-up.

Comparison Methods	1	2	3	4	5	6	7	8	9	10	11	12	All	Imp.
PerAct	5.3	66.7	0.0	84.0	38.7	13.3	16.0	2.7	41.3	0.0	97.3	6.7	31.0	⇑ 12.6
GNFactor	2.7	92.0	1.3	84.0	80.0	28.0	8.0	1.3	46.7	32.0	100	0.0	40.0	⇑ 3.6
Fine-Tuning	8.0	18.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**68.0**	**100**	10.7	17.1	⇑ 25.6
ER	10.7	58.7	**24.0**	**81.3**	**76.0**	**12.0**	5.3	4.0	**78.7**	8.0	89.3	1.3	37.4	⇑ 6.2
Ours-w/oSEP&SRD	**13.3**	**74.7**	**28.0**	76.0	52.0	8.0	**9.3**	**12.0**	53.3	**25.3**	96.0	2.7	37.6	⇑ 6.0
Ours-w/oSRD	8.0	**78.7**	6.7	**81.3**	60.0	**12.0**	**9.3**	**20.0**	52.0	9.3	90.7	**25.3**	**37.8**	⇑ 5.8
Ours	**17.3**	**74.7**	9.3	**82.7**	**62.7**	**22.7**	**22.7**	9.3	**85.3**	9.3	**98.7**	**28.0**	**43.6**	–

Table 4. Comparison results in terms of success rate (%) on Living Room dataset when setting the various size of memory buff $\mathcal{M}$.

Buffer size	**6-3** (2 steps)					**6-2** (4 steps)				
	1-6	7-12	All	Avg.	For.	1-6	7-12	All	Avg.	For.
$\lvert\mathcal{M}\rvert = 2$	42.0	41.3	41.7	44.9	**4.4**	32.0	30.7	31.3	37.2	18.8
$\lvert\mathcal{M}\rvert = 4$	**44.9**	**42.2**	**43.6**	**47.6**	7.7	**45.8**	**35.3**	**40.6**	**43.5**	**10.9**
$\lvert\mathcal{M}\rvert = 6$	**50.0**	**36.7**	**43.3**	**45.0**	**2.2**	**50.0**	**34.7**	**42.3**	**45.4**	**8.0**

success score after the last step for the base task (Base) $\mathcal{I}_B^M$, incremental tasks (Novel) $\mathcal{I}_N^M$ and all manipulation skills (All) $\mathcal{I}_A^M$. These metrics respectively reflect the robustness of old skill forgetting , the capacity of novel skill learning, as well as its overall performance. Additionally, we introduce a Avg. metric A and For. metric F to measure average performance and skill-wise forgetting rate over the whole NERL process, where $A = \frac{1}{M}\sum_{m=1}^{M}\mathcal{I}_A^m$ and $F = \frac{1}{N^m}\sum_{i=1}^{N^m}\max_{m\in\{1,\dots,M-1\}}(\mathcal{I}_i^m - \mathcal{I}_i^M)$.

4.2. Comparison Performance

We present comparison results between our NBCagent and other methods on Kitchen and Living Room datasets in Tabs. 1, 2 and 3. As shown in Tab. 1, NBCagent significantly outperforms compared methods by $1.2\% \sim 51.2\%$ in terms of success score on base task and $5.1\% \sim 17.6\%$ on incremental tasks. This indicates that NBCagent can learn skill-specific and skill-shared knowledge, thereby effectively addressing old skill forgetting and novel skill learning. Furthermore, as presented in Tabs. 2 and 3, our model exhibits the highest mean success rate across all manipulation skills, improving by $1.2\% \sim 36.3\%$ and $3.6\% \sim 25.6\%$ respectively when compared to other methods. This demonstrates the effectiveness of our model in addressing the NERL problem. Additionally, our NBCagent achieves a large improvements about $2.5\% \sim 41.4\%$ and $0.1\% \sim 36.4\%$ in terms of Avg. and For. metrics, which suggests the robust and significant performance of NBCagent over the whole NERL process. Surprisingly, as shown in Tabs. 2 and 3, our NBCagent performs better than joint training methods, specifically PerAct and GNFactor, providing additional evidence to support the efficacy of our model. The comparison results under other settings can be found in **Appendix C**.

4.3. Ablation Studies

To evaluate effectiveness of each module in our NBCagent, we eliminate them one by one and present results in Tabs. 1, 2 and 3. Compared to Ours, the scores of Ours-w/oSRD on both base task and all tasks are dropped by $3.8\% \sim 12.5\%$ and $0.8\% \sim 5.8\%$ respectively. This indicates that SRD can effectively learn skill-shared knowledge to tackle old task forgetting. In addition, Ours-w/oSRD outperforms Ours-w/oSEP&SRD on incremental tasks by $1.3\% \sim 11.4\%$, which demonstrates that SEP benefits our model in learning novel skills by performing skill-specific knowledge learning. Furthermore, to evaluate the effectiveness of SSR module, we visualize the rendering results in Fig. 4. It suggests that our SSR module can efficiently complete skill-shared semantics under the supervision of novel view and diffusion features, thereby achieving an improvement about $0.5\% \sim 3.1\%$ in terms of Avg. compared with ER. The visualization results in Fig. 3 also shows the effectiveness of our model to tackle the NERL problem. We also explore the impact of various sizes of memory buffer $\mathcal{M}$. As shown in Tab. 4, with $|\mathcal{M}| = 6$, the forgetting rate of our model notably reduced by $1.1\% \sim 10.8\%$ in comparison to $|\mathcal{M}| = 4$ and 2, respectively. This indicates that increasing the memory size significantly addresses catastrophic forgetting but also incurs a larger memory load.

5. Conclusion

In this paper, we explore a pioneering Never-ending Embodied Robot Learning (NERL) problem and propose a novel NBCagent to continually learn skill-wise knowledge. Specifically, we propose a skill-specific evolving planner to decouple the skill-wise knowledge to effectively learning novel skills. In addition, we design a skill-shared semantics rendering module and skill-shared representation distillation module to tackle catastrophic forgetting on old skills from semantics and representation aspects. We develop two NERL benchmarks and expensive experiments on them verify the effectiveness of our NBCagent against baselines.

References

Ayub, A. and Fendley, C. Few-shot continual active learning by a robot. In *NeurIPS*, volume 35, pp. 30612–30624, 2022.

Ayub, A. and Wagner, A. R. Cbcl-pr: A cognitively inspired model for class-incremental learning in robotics. *IEEE Transactions on Cognitive and Developmental Systems*, 2023.

Ayub, A., De Francesco, Z., Holthaus, P., Nehaniv, C. L., and Dautenhahn, K. Continual learning through human-robot interaction–human perceptions of a continual learning robot in repeated interactions. *arXiv preprint arXiv:2305.16332*, 2023.

Bang, J., Kim, H., Yoo, Y., Ha, J.-W., and Choi, J. Rainbow memory: Continual learning with a memory of diverse samples. In *CVPR*, pp. 8218–8227, 2021.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

Brohan, A., Chebotar, Y., Finn, C., Hausman, K., Herzog, A., Ho, D., Ibarz, J., Irpan, A., Jang, E., Julian, R., et al. Do as i can, not as i say: Grounding language in robotic affordances. In *CoRL*, pp. 287–318. PMLR, 2023.

Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P. K., Torr, P. H., and Ranzato, M. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Derakhshani, M. M., Zhen, X., Shao, L., and Snoek, C. Kernel continual learning. In *ICML*, pp. 2621–2631. PMLR, 2021.

Dong, J., Liang, W., Cong, Y., and Sun, G. Heterogeneous forgetting compensation for class-incremental learning. In *ICCV*, pp. 11742–11751, 2023.

Douillard, A., Cord, M., Ollion, C., Robert, T., and Valle, E. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, pp. 86–102. Springer, 2020.

Driess, D., Schubert, I., Florence, P., Li, Y., and Toussaint, M. Reinforcement learning with neural radiance fields. In *NeurIPS*, volume 35, pp. 16931–16945, 2022.

Driess, D., Xia, F., Sajjadi, M. S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Gao, C., Gao, H., Guo, S., Zhang, T., and Chen, F. Cril: Continual robot imitation learning via generative and prediction model. In *IROS*, pp. 6747–5754. IEEE, 2021.

Goyal, A., Mousavian, A., Paxton, C., Chao, Y.-W., Okorn, B., Deng, J., and Fox, D. Ifor: Iterative flow minimization for robotic object rearrangement. In *CVPR*, pp. 14787–14797, 2022.

Goyal, A., Xu, J., Guo, Y., Blukis, V., Chao, Y.-W., and Fox, D. Rvt: Robotic view transformer for 3d object manipulation. *arXiv preprint arXiv:2306.14896*, 2023.

Hafez, M. B. and Wermter, S. Continual robot learning using self-supervised task inference. *IEEE Transactions on Cognitive and Developmental Systems*, 2023.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

Jaegle, A., Borgeaud, S., Alayrac, J.-B., Doersch, C., Ionescu, C., Ding, D., Koppula, S., Zoran, D., Brock, A., Shelhamer, E., et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021.

James, S., Ma, Z., Arrojo, D. R., and Davison, A. J. Rlbench: The robot learning benchmark & learning environment.

IEEE Robotics and Automation Letters, 5(2):3019–3026, 2020.

James, S., Wada, K., Laidlow, T., and Davison, A. J. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *CVPR*, pp. 13739–13748, 2022.

Jiang, Y., Gupta, A., Zhang, Z., Wang, G., Dou, Y., Chen, Y., Fei-Fei, L., Anandkumar, A., Zhu, Y., and Fan, L. Vima: General robot manipulation with multimodal prompts. *arXiv*, 2022.

Jung, S., Ahn, H., Cha, S., and Moon, T. Continual learning with node-importance based adaptive group sparse regularization. In *NeurIPS*, volume 33, pp. 3647–3658, 2020.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *ICML*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Li, A. C., Prabhudesai, M., Duggal, S., Brown, E., and Pathak, D. Your diffusion model is secretly a zero-shot classifier. In *ICCV*, pp. 2206–2217, October 2023.

Li, G., Zhai, Y., Chen, Q., Gao, X., Zhang, J., and Zhang, Y. Continual few-shot intent detection. In *COLING*, pp. 333–343, 2022.

Li, Z. and Hoiem, D. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *ICML*, pp. 8748–8763. PMLR, 2021.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. icarl: Incremental classifier and representation learning. In *CVPR*, pp. 2001–2010, 2017.

Rohmer, E., Singh, S. P., and Freese, M. V-rep: A versatile and scalable robot simulation framework. In *IROS*, pp. 1321–1326. IEEE, 2013.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *CVPR*, pp. 10684–10695, 2022.

Shah, D., Osiński, B., Levine, S., et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *CoRL*, pp. 492–504. PMLR, 2023.

Shim, D., Lee, S., and Kim, H. J. Snerl: Semantic-aware neural radiance fields for reinforcement learning. In *ICML*. PMLR, 2023.

Shridhar, M., Manuelli, L., and Fox, D. Cliport: What and where pathways for robotic manipulation. In *CoRL*, pp. 894–906. PMLR, 2022.

Shridhar, M., Manuelli, L., and Fox, D. Perceiver-actor: A multi-task transformer for robotic manipulation. In *CoRL*, pp. 785–799. PMLR, 2023.

Sun, G., Liang, W., Dong, J., Li, J., Ding, Z., and Cong, Y. Create your world: Lifelong text-to-image diffusion. *arXiv preprint arXiv:2309.04430*, 2023.

Sun, Q., Lyu, F., Shang, F., Feng, W., and Wan, L. Exploring example influence in continual learning. In *NeurIPS*, volume 35, pp. 27075–27086, 2022.

Tiwari, R., Killamsetty, K., Iyer, R., and Shenoy, P. Gcr: Gradient coreset based replay buffer selection for continual learning. In *CVPR*, pp. 99–108, 2022.

Toldo, M. and Ozay, M. Bring evanescent representations to life in lifelong class incremental learning. In *CVPR*, pp. 16732–16741, 2022.

Wan, W., Zhu, Y., Shah, R., and Zhu, Y. Lotus: Continual imitation learning for robot manipulation through unsupervised skill discovery. *arXiv preprint arXiv:2311.02058*, 2023.

Wang, L., Zhang, X., Li, Q., Zhu, J., and Zhong, Y. Coscl: Cooperation of small continual learners is stronger than a big one. In *ECCV*, pp. 254–271. Springer, 2022.

Wang, L., Zhang, X., Su, H., and Zhu, J. A comprehensive survey of continual learning: Theory, method and application. *arXiv preprint arXiv:2302.00487*, 2023.

Wu, Z., Baek, C., You, C., and Ma, Y. Incremental learning via rate reduction. In *CVPR*, pp. 1125–1133, 2021.

Xiang, Y., Fu, Y., Ji, P., and Huang, H. Incremental learning using conditional adversarial networks. In *ICCV*, pp. 6619–6628, 2019.

Xu, J., Liu, S., Vahdat, A., Byeon, W., Wang, X., and De Mello, S. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *CVPR*, pp. 2955–2966, 2023.

Yang, B., Deng, X., Shi, H., Li, C., Zhang, G., Xu, H., Zhao, S., Lin, L., and Liang, X. Continual object detection via prototypical task correlation guided gating mechanism. In *CVPR*, pp. 9255–9264, 2022.

You, Y., Li, J., Reddi, S., Hseu, J., Kumar, S., Bhojanapalli, S., Song, X., Demmel, J., Keutzer, K., and Hsieh, C.-J. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.

Yu, A., Ye, V., Tancik, M., and Kanazawa, A. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, pp. 4578–4587, 2021.

Ze, Y., Yan, G., Wu, Y.-H., Macaluso, A., Ge, Y., Ye, J., Hansen, N., Li, L. E., and Wang, X. Gnfactor: Multi-task real robot learning with generalizable neural feature fields. In *CoRL*, pp. 284–301. PMLR, 2023.

Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, pp. 2165–2183. PMLR, 2023.

A. Dataset Description

In order to emulate the operational environments of domestic robots, we design two NERL benchmark datasets using RLBench, called Kitchen and Living Room. In particular, we assemble Kitchen by gathering 10 skills that are pertinent to kitchen settings, and we construct Living Room, which includes 12 skills that are related to living room situations. Each skill includes a training set of 20 episodes and a test set of 25 episodes. We provide examples of brief overviews of these skills and the number of keyframes for each skill in Tab. 5. In addition, the **Skill No.** represents the corresponding skill order mentioned in the Experiment section. The types of skill variables include: the color, position, and size of the target object. The skill description may randomly sample from twenty colors: red, maroon, lime, green, blue, navy, yellow, cyan, magenta, silver, gray, orange, olive, purple, teal, azure, violet, rose, black, and white; and two sizes: short and tall. The position variable is skill-specific. For example, in the skill 'place wine at rack location', the target object 'rack' has three positions: 'middle', 'left', 'right'.

Table 5. **Skill Details in Kitchen & Living Room**

Skill No.	Skill in Kitchen	Language Example	Avg. Keyframes
1	close microwave	"close microwave"	2.3
2	meat off grill	"take the steak off the grill"	6.0
3	open grill	"open the grill"	4.5
4	open wine bottle	"open wine bottle"	3.3
5	pick up cup	"pick up the red cup"	3.8
6	turn tap	"turn left tap"	3.0
7	place wine at rack location	"stack the wine bottle to the left of the rack"	6.1
8	put knife on chopping board	"put the knife on the chopping board"	5.1
9	stack wine	"stack wine bottle"	7.4
10	take plate off colored dish rack	"take plate off the red colored rack"	6.3
Skill No.	**Skill in Living Room**	**Language Example**	**Avg. Keyframes**
1	close door	"close the door"	4.8
2	close laptop lid	"close laptop lid"	6.0
3	hang frame on hanger	"hang frame on hanger"	5.3
4	lamp on	"turn on the light"	3.5
5	open drawer	"open the bottom drawer"	4.8
6	open window	"open left window"	6.0
7	push buttons	"push the white button"	4.8
8	put item in drawer	"put the item in the top drawer"	14.3
9	put rubbish in bin	"put rubbish in bin"	5.0
10	sweep to dustpan of size	"sweep dirt to the tall dustpan"	6.2
11	take usb out of computer	"take usb out of computer"	3.3
12	water plants	"pour some water on the plant"	6.3

Table 6. Comparison results on Kitchen under the setting of **5-5**. **Red** and **Blue** represents the highest results and runner-up.

Comparison Methods	1	2	3	4	5	6	7	8	9	10	All.	Imp.
PerAct	96.0	77.3	53.3	30.7	37.3	52.0	2.7	18.7	4.0	73.3	44.5	⇑ 0.4
GNFactor	92.0	60.0	70.7	12.0	46.7	52.0	5.3	22.7	10.7	70.7	44.3	⇑ 0.6
Fine-Tuning	10.7	0.0	33.3	33.3	1.3	49.3	**6.7**	14.7	5.3	56.0	21.1	⇑ 23.8
ER	**96.0**	72.0	**56.0**	18.7	37.3	45.3	0.0	14.7	1.3	66.7	40.8	⇑ 4.1
Ours-w/oSEP&SRD	86.7	**74.7**	**64.0**	28.0	26.7	52.0	**6.7**	17.3	**8.0**	49.3	**41.3**	⇑ 3.6
Ours-w/oSRD	8.0	**81.3**	36.0	**38.7**	**41.3**	**62.7**	**4.0**	**46.7**	**9.3**	**68.0**	39.6	⇑ 5.3
Ours	**94.7**	52.0	38.7	**44.0**	**38.7**	**64.0**	2.7	**34.7**	**9.3**	**70.7**	**44.9**	–

Table 7. Comparison results on Living Room under the setting of **6-2**. **Red** and **Blue** represents the highest results and runner-up.

Comparison Methods	1	2	3	4	5	6	7	8	9	10	11	12	All.	Imp.
PerAct	5.3	66.7	0.0	84.0	38.7	13.3	16.0	2.7	41.3	0.0	97.3	6.7	31.0	⇑ 9.6
GNFactor	2.7	92.0	1.3	84.0	80.0	28.0	8.0	1.3	46.7	32.0	100	0.0	40.0	⇑ 0.6
Fine-Tuning	0.0	37.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	96.0	**18.7**	12.7	⇑ 27.9
ER	4.0	**89.3**	**17.3**	80.0	38.7	14.7	**20.0**	2.7	60.0	**4.0**	90.7	4.0	35.4	⇑ 5.2
Ours-w/oSEP&SRD	2.7	**82.7**	14.7	76.0	30.7	**24.0**	18.7	9.3	76.0	**4.0**	94.7	**10.7**	37.0	⇑ 3.6
Ours-w/oSRD	**12.0**	65.3	14.7	**82.7**	**49.3**	18.7	**28.0**	**18.7**	**81.3**	**6.7**	**100.0**	0.0	**39.8**	⇑ 0.8
Ours	**8.0**	61.3	**32.0**	**84.0**	**58.7**	**30.7**	4.0	**17.3**	**80.0**	**6.7**	**98.7**	5.3	**40.6**	–

B. Training Details

As shown in Tab. 8, we give the hyper-parameters used in NBCagent. The overall learning rate of the training process is 0.0005, optimized by the LAMB algorithm. All methods employ the same methodology to construct a voxel grid of size 100^3. During the NeRF training, we utilize an additional 19 camera perspectives to furnish supervisory information. For NERL, we conduct two NERL settings in each of the two scenarios, which are 5-1, 5-1 in Kitchen, and 6-3, 6-2 in Living Room. In SEP, NBCagent detects novel skills through the cosine similarity of the skill language description feature vectors. We set the judgment threshold δ as 0.8 for our experiments.

Table 8. **Hyper-parameters** used in NBCagent

Variable Name	Value
image size	$128 \times 128 \times 3$
input voxel size	$100 \times 100 \times 1000$
batch size B	2
optimizer	LAMB
learning rate	0.0005
number of transformer blocks	6
number of sampled points for NERagent	64
number of latents in Perceiver Transformer	2046
dimension of Stable Diffusion features	512
dimension of CLIP language features	512
hidden dimension of NeRF blocks	512
size of LoRA layers N^r	10
base task iterations	80K
incremental task iterations	20k
SSR loss weight λ_1	0.1
SSR loss function $\mathcal{L}_s$	MSE-loss
SRD loss weight λ_2	0.2
SRD loss function $\mathcal{L}_{srd}$	KL divergence
SRD temperature $\mathcal{T}$	3
novel skill threshold δ	0.8
size of memory buffer $\mathcal{M}$	4
number of base skill in living room	6
number of incremental skill in living room	3, 2
number of base skill in kitchen	5
number of incremental skill in kitchen	1, 5

C. More Comparison Experiments

Tabs. 6 and 7 show the detailed scores of each model in two scenarios under the NERL settings of 5-5 and 6-2.With the configuration of memory buffer size $|\mathcal{M}| = 4$, the average score of NBCagent across all skills still surpasses other models, approximately between $0.4\% \sim 23.8\%$ and $0.6\% \sim 27.9\%$. This indicates that NBCagent performs the best in solving NERL problems. Fig. 5 shows the visualization results of more skills among various models. We observe that some skills are not difficult to learn the general actions, but when the skill requires the target object to be in a specific location, the agent has difficulty completing it accurately; for example, 'open the drawer' is not difficult, but when the requirement is 'open the **top** drawer', the performance of NBCagent is better than other models.

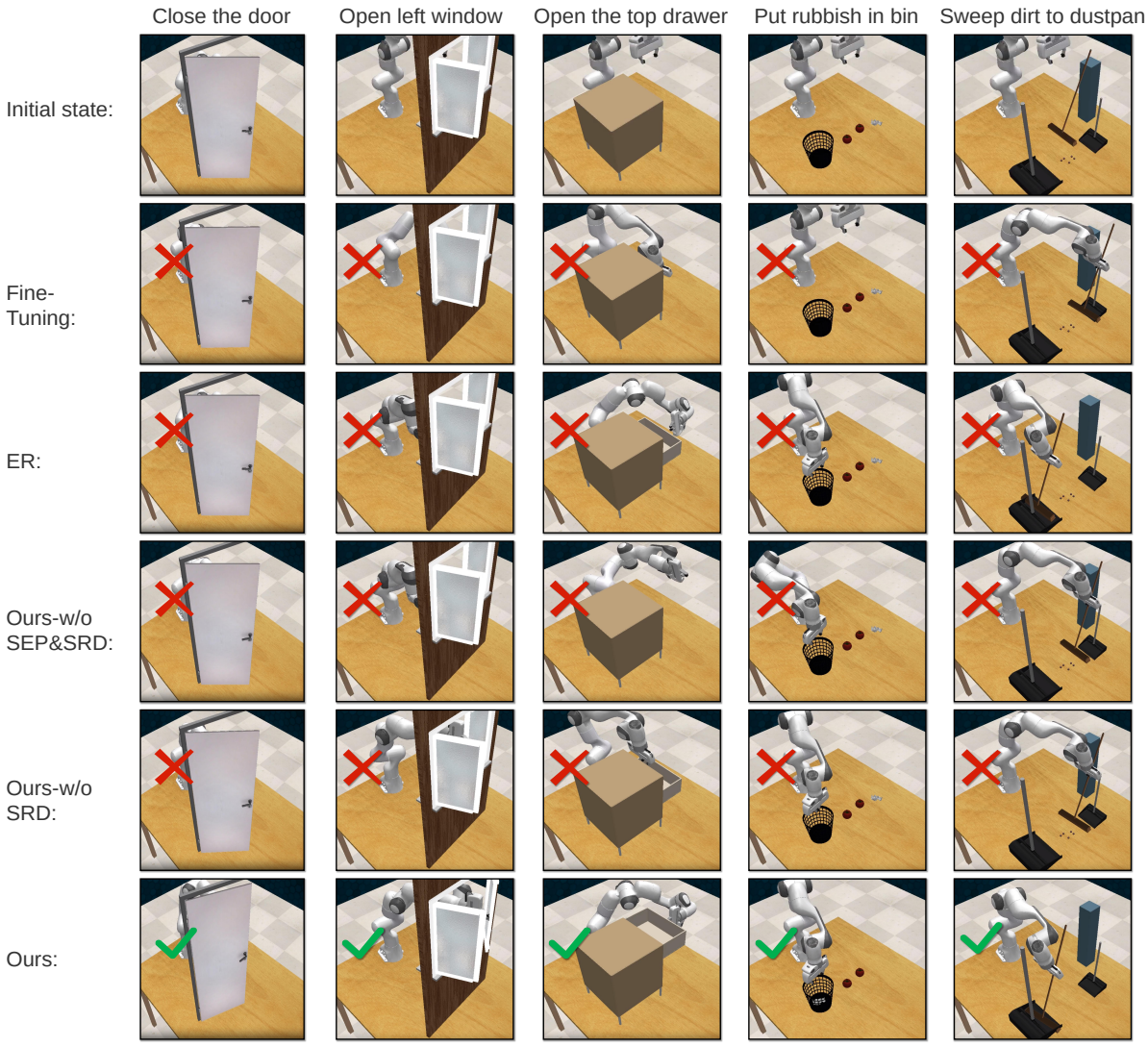

Figure 5. Visualization of prediction results on various skills between Ours, Ours-w/oSRD, Ours-w/oSEP&SRD, ER and Fine-Tuing.