# Pretrained Bayesian Non-parametric Knowledge Prior in Robotic Long-Horizon Reinforcement Learning

Yuan Meng[1], Xiangtong Yao[1], Kejia Chen[1], Yansong Wu[1], Liding Zhang[1],
Zhenshan Bing[2,†], and Alois Knoll[1] *IEEE fellow*

*Abstract*— **Reinforcement learning (RL) methods typically learn new tasks from scratch, often disregarding prior knowledge that could accelerate the learning process. While some methods incorporate previously learned skills, they usually rely on a fixed structure, such as a single Gaussian distribution, to define skill priors. This rigid assumption can restrict the diversity and flexibility of skills, particularly in complex, long-horizon tasks. In this work, we introduce a method that models potential primitive skill motions as having non-parametric properties with an unknown number of underlying features. We utilize a Bayesian non-parametric model, specifically Dirichlet Process Mixtures, enhanced with birth and merge heuristics, to pre-train a skill prior that effectively captures the diverse nature of skills. Additionally, the learned skills are explicitly trackable within the prior space, enhancing interpretability and control. By integrating this flexible skill prior into an RL framework, our approach surpasses existing methods in long-horizon manipulation tasks, enabling more efficient skill transfer and task success in complex environments. Our findings show that a richer, non-parametric representation of skill priors significantly improves both the learning and execution of challenging robotic tasks. All data, code, and videos are available at `https://ghiara.github.io/HELIOS/`.**

## I. INTRODUCTION

Advances in artificial intelligence have shown that utilizing prior experience can lead to more efficient solutions for new tasks. However, in reinforcement learning, agents often start each new task from scratch, requiring extensive data collection and computational resources, especially in real-world scenarios. To address this, recent work has explored extracting "skills" or "knowledge" from unstructured experience. These temporally extended actions represent useful behaviors that can be transferred across tasks. Previous approaches build skill datasets from human demonstrations or autonomous exploration, then use these skills in downstream learning

by training a high-level policy to select from the skill datasets. However, these methods often assume a fixed structure, such as a single Gaussian distribution, to represent skill priors [1]. This rigid assumption restricts the diversity and adaptability of skills, especially in complex, long-horizon tasks. Moreover, as the number of skills grows, exploring the skill space becomes computationally intensive, restricting learning efficiency.

In this work, we propose a more flexible approach to model skill priors. Unlike previous methods, we assume that primitive skill motions exhibit non-parametric properties [2] with an unknown number of underlying features. By leveraging a Bayesian non-parametric model, specifically Dirichlet Process Mixtures (DPM) [3], combined with a more advanced Gated Recurrent Unit (GRU) based skill prior module [4], we allow the skill prior to capture a wider range of potential behaviors with a dynamic adaptive structure. Additionally, we incorporate birth and merge heuristics [3] to dynamically adjust the skill prior, making the captured skills both diverse and explicitly trackable within the prior space, enhancing interpretability. We integrate this non-parametric skill prior into an RL framework and demonstrate that it significantly improves performance in long-horizon manipulation tasks. Our approach not only enables efficient skill transfer in complex environments but also shows that a non-parametric representation of skill priors enhances both learning and task execution. To validate our method, we test it on the Franka-Kitchen Benchmark [5], where our model outperforms baseline methods in task success and adaptability.

Our main contributions are: (1) Introducing a Bayesian non-parametric model for skill priors, providing a flexible and interpretable skill representation, (2) integrating this skill prior module into an RL framework for improved long-horizon manipulation, and (3) demonstrating through empirical results that our method outperforms recent approaches in complex, multi-goal robotic manipulation tasks. We name our proposed framework **HELIOS**: **H**ierarchical **E**ncoding of **L**ong-horizon **I**nference **O**ff-policy Reinforcement Learning

[1]Yuan Meng, Xiangtong Yao, Kejia Chen, Yansong Wu, Liding Zhang, and Alois Knoll are with the School of Computation, Information and Technology, Technical University of Munich, Germany.

[2]Zhenshan Bing is with the State Key Laboratory for Novel Software Technology, Nanjing University, China.

† Corresponding author: `zhenshan.bing@tum.de`

Framework with Bayesian Non-parametric Skill Prior.

## II. RELATED WORK

**Meta reinforcement learning** [6], [7], [8] (meta-RL) seeks to extract useful priors from past experiences to improve learning efficiency on new tasks. While meta-RL approaches share the goal of leveraging prior knowledge, they typically require a predefined set of training tasks and rely on online data collection during pre-training. This requirement limits their applicability in leveraging large offline datasets. Several recent works address offline RL [9], [10], [11], [12] by learning directly from logged agent experience, with no need for additional environment interactions. However, it typically requires reward annotations specific to the target task, which is challenging to obtain for large-scale datasets gathered from diverse tasks. In contrast, our framework learns skills offline from unstructured action patterns and bypasses the need for reward-labeled data, enabling flexible and scalable transfer across various downstream tasks.

**Pretrained behavior priors** are commonly used in offline RL to guide policy learning, helping to mitigate value overestimation for actions outside the training distribution. Recent studies have applied priors to either individual actions [13] or temporally extended skills [1], incorporating offline experience to support downstream task learning. Building on this, our work employs priors over temporally extended skills, specifically for long-horizon inference.

**Bayesian (non-)parametric models** are essential for managing uncertainty and capturing complex distributions in robotic learning. In Bayesian parametric models, the number of parameters is fixed and predefined, typically representing distributions through a finite set of parameters. Common examples include Gaussian Mixture Models (GMM) [14], such as in TIGR [12], which employs GMM within a variational autoencoder (VAE) to capture multimodal RL task distributions, allowing efficient adaptation to nonstationary and nonparametric robotic environments. In contrast, Bayesian non-parametric models do not require a predefined number of parameters. Instead, they adapt their complexity based on the observations, enabling an infinite-dimensional parameter space. This adaptability is particularly useful in robotics, where features in skill space and behavioral patterns can be dynamic or unknown. Common non-parametric models include Dirichlet Process Mixtures [3], the Stick-Breaking Process (SB) [15], and the Chinese Restaurant Process (CRP) [16]. While these models are well-established in theoretical machine learning and computer vision, their application to the complexities of robotic learning needs further exploration.

## III. PRELIMINARIES

In this section, we first introduce the memorized online variational inference method for Dirichlet Process Mixtures, followed by the integration of maximum entropy principles in downstream off-policy RL learning.

### A. Variational Inference of DPM

DPM is a widely used Bayesian non-parametric model designed to capture an infinite mixture of clusters when modeling a set of observations $x = x_{1:N}$. The generative process of the DPM defines how clusters and data points are formed under a Bayesian non-parametric framework. In this model, each component $\theta_k^*$ is a cluster parameter sampled from a base distribution $\mathscr{H}(\lambda)$, parameterized by $\lambda$. To determine the mixture proportions of these clusters, a mixing proportion distribution $\pi$ is drawn from the Generalized Ewens Distribution (GEM) with concentration parameter $\alpha$, which specifies the probability of assigning new data points to existing clusters versus creating new ones. Given the $\pi$, a cluster assignment variable $c_i$ is sampled for each data point $x_i$ from a categorical distribution $\text{Cat}(\pi)$, where $c_i = k$ indicates that $x_i$ is assigned to cluster $k$. Finally, the data point $x_i$ itself is generated from the distribution $\mathscr{F}(\theta_i)$, parametrized by parameterized by the corresponding cluster parameter $\theta_{c_i}^*$. This process can be expressed as:

$$
\begin{aligned}
\theta_k^*|\lambda &\sim \mathscr{H}(\lambda), \\
\pi|\alpha &\sim \text{GEM}(\alpha), \\
c_i|\pi &\sim \text{Cat}(\pi), \\
x_i|c_i &\sim \mathscr{F}(\theta_{c_i}^*).
\end{aligned}
\tag{1}
$$

In this work, we use an online variational inference-based method to estimate the true posterior density of observations, as it provides faster and more scalable solutions than sampling-based approaches. Given Eq.(1), the joint probability of the DPM parameters is expressed as:

$$
\begin{aligned}
p(x,c,\theta,\beta) = &\prod_{n=1}^{N} \mathscr{F}(x_n|\theta_{c_n})\text{Cat}(c_n|\pi(\beta)) \\
&\prod_{k=1}^{\infty} \mathscr{B}(\beta_k|1,\alpha)\mathscr{H}(\theta_k|\lambda).
\end{aligned}
\tag{2}
$$

Here, the parameter $\beta$ of mixing proportion $\pi$ is sampled from a beta distribution parameterized by $\alpha$. Since we cannot obtain the exact posterior $p(c,\theta,\beta|x)$, the goal in variational inference is to find an optimal variational distribution $q^*(c,\theta,\beta)$ that minimizes the Kullback–Leibler (KL) divergence from the true posterior.

This turns out to maximize the evidence lower bound (ELBO), which can be derived as follows:

$$\text{ELBO}(q) = \mathbb{E}[\log p(\boldsymbol{x}|\boldsymbol{c}, \boldsymbol{\theta}, \boldsymbol{\beta})] \\ - \mathbb{KL}(q(\boldsymbol{c}, \boldsymbol{\theta}, \boldsymbol{\beta})||p(\boldsymbol{c}, \boldsymbol{\theta}, \boldsymbol{\beta})). \quad (3)$$

For the DPM, we define the variational distribution $q$ using the mean field assumption [17], where each latent variable has its own independent variational factor (indicated by $\hat{*}$). Specifically, we obtain:

$$q(\boldsymbol{c}, \boldsymbol{\theta}, \boldsymbol{\beta}) = \prod_{n=1}^{N} q(c_n|\hat{r}_n) \prod_{k=1}^{K} q(\beta_k|\hat{\alpha}_{k_1}, \hat{\alpha}_{k_0}) q(\theta_k|\hat{\lambda}_k),$$
$$= \prod_{n=1}^{N} \underbrace{\text{Cat}(c_n|\hat{r}_{n_1:n_K})}_{q_{c_n}} \prod_{k=1}^{K} \underbrace{\mathscr{B}(\beta_k|\hat{\alpha}_{k_1}, \hat{\alpha}_{k_0})}_{q_{\beta_k}} \underbrace{\mathscr{H}(\theta_k|\hat{\lambda}_k)}_{q_{\theta_k}},$$
$$(4)$$

where $q_{c_n}$, $q_{\beta_k}$, and $q_{\theta_k}$ are categorical factors, beta distribution factors, and base distribution factors, respectively. Since the true posterior is infinite, we make the $q$ tractable by truncating the assignment factor to $K$ components, setting $q(c_n = k) = 0$ for $k > K$. This allows inference to focus on a finite set of $K$ components, approximating the true infinite posterior effectively when $K$ is large enough. In the special case where both $\mathscr{H}$ and $\mathscr{F}$ belong to the exponential family, the optimization objective for the ELBO in Eq.(3) can be reformulated in terms of the expected mass $\hat{N}_k$ and the expected sufficient statistics $s_k(x)$ for each component $k$ [3]:

$$\text{ELBO}(q) = \sum_{k=1}^{K} \left[ \mathbb{E}_q[\theta_k]^\top s_k(x) - \hat{N}_k[a(\theta_k)] + \hat{N}_k[\log \pi_k(\beta)] \right. \\ + \mathbb{E}_q[\log \frac{\mathscr{B}(\beta_k|1, \alpha)}{q(\beta_k|\hat{\alpha}_{k_1}, \hat{\alpha}_{k_0})}] + \mathbb{E}_q[\log \frac{\mathscr{H}(\theta_k|\lambda)}{q(\theta_k|\hat{\lambda}_k)}] \\ \left. - \sum_{n=1}^{N} \hat{r}_{nk} \log \hat{r}_{nk} \right].$$
$$(5)$$

We use a coordinate ascent approach to iteratively update each variational factor. First, we update the local parameters $\hat{r}_{nk}$ for each cluster assignment factor $q_{c_n}$. Then, we adjust the global parameters in $q_{\beta_k}$ and $q_{\theta_k}$ to maximize the ELBO [3].

In this work, we apply an online heuristic method, MemoVB, which introduces birth and merge moves for dynamic cluster adjustments [3]. To create new clusters, we collect subsamples $x'$ that are poorly described by a single cluster in each batch and fit a separate DPM with $K'$ initial clusters. Assuming the number of active clusters before the birth move is $K$, we decide to accept or reject new cluster proposals by comparing the likelihood of assigning $x'$ to $K + K'$ clusters versus $K$ clusters alone. To complement the birth move, the merge move

consolidates two clusters into one if merging improves the ELBO objective, reducing the number of clusters to $K - 1$.

### B. Maximum Entropy in RL

In offline RL, the maximum entropy principle promotes exploration by encouraging agents to maximize both the expected return $r$ and the entropy $H$ of their policy [11]. The objective is formulated as:

$$J(\varphi) = \sum_{t=1}^{T} \mathbb{E}_{(s_0, a_0) \sim \rho_\pi} \left[ r(s_t, a_t) + \omega H(\pi_\varphi(\cdot|s_t)) \right], \quad (6)$$

where $r(s_t, a_t)$ is the reward for the state-action pair, $\rho_\pi$ is the state-action distribution induced by policy $\pi_\varphi$ (parameterized by $\varphi$), $\omega$ is the temperature parameter that balances the trade-off between reward and entropy. This approach is particularly advantageous in offline settings, where the agent's exploration is constrained to a large offline dataset $\mathbb{D}$. Balancing exploration and exploitation is challenging in such settings, especially as the number of embedded skills increases rapidly.

## IV. METHOD

In this section, we present the methodology of our framework, as illustrated in Fig. 1. The framework is organized into two main phases: skill prior pretraining and RL for long-horizon manipulation. We start with the problem formulation in the RL setting, followed by a detailed explanation of the Bayesian non-parametric skill prior to pretraining (Phase I), and conclude with the downstream training process for long-horizon manipulation tasks (Phase II) using the pre-trained skill prior.

### A. Problem Statement

In this work, we address the challenging problem of long-horizon manipulation in reinforcement learning by developing a framework that utilizes Bayesian non-parametric skill priors to guide downstream RL policy learning. We formulate this problem within the standard RL framework, represented as a Markov Decision Process (MDP) defined by the tuple $\{S, A, T, R, \gamma\}$, where $S$ is the state space, $A$ the action space, $T$ the transition function, $R$ the reward function, and $\gamma$ the discount factor. Our approach is motivated by the need for efficient long-horizon task learning within a data-driven deep RL setting, where conventional RL methods often struggle due to the complexity of learning from scratch. Inspired by prior work [1], we recognize that existing unstructured datasets contain informative data that can be captured by a skill prior module to accelerate exploration in downstream tasks. In this context, we assume that action-based skills within the dataset exhibit an
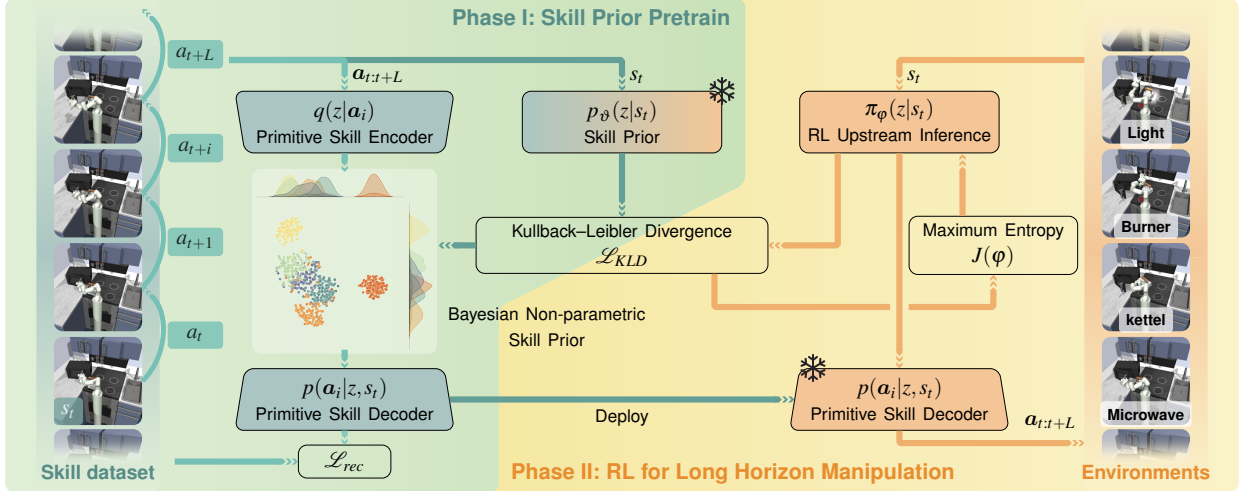
Fig. 1. HELIOS Framework overview. The training process is divided into two phases. In Phase I, a VAE with GRU modules is used to pre-train a skill representation model from a dataset of action trajectories. The model leverages a DPM to capture the non-parametric nature of skill priors, aiding in learning precise action patterns and subsequent effective task representations. In Phase II, this pre-trained skill decoder and prior are deployed within a RL framework to address long-horizon manipulation tasks. Here, the upstream inference model uses soft actor-critic structure to learn specific task reasoning, ensuring the successful execution of complex, extended long-horizon tasks.

unknown number of features, embodying non-parametric properties [2] that cannot be adequately represented by a single Gaussian distribution. To address this, we employ a DPM for skill prior learning, allowing our model to capture flexible, non-parametric skill representations. The pre-trained skill prior is then integrated into the RL framework, enhancing both exploration efficiency and adaptability in complex, long-horizon tasks.

### B. Bayesian Non-parametric Skill Prior

The training process is divided into two phases. As shown in Fig.1, the first phase (highlighted in green) focuses on training a Bayesian non-parametric skill prior using a DPM model in the latent representation space. This phase leverages a VAE with a GRU-based backbone to model the non-parametric distributions of temporally extended skills based on an unstructured dataset.

During this phase, training proceeds iteratively. First, we freeze the network parameters to update the DPM. In the initial epoch, we start with a single component in the DPM to facilitate computation. In the subsequent epochs, we collect embeddings of the inferred skill posterior $q(z|a_i)$ and fit them to the DPM from the previous epoch. According to the DPM optimization objective from Eq.(5), new components can be created or existing ones merged to improve the ELBO. The latest DPM configuration replaces the previous one, preparing for the next iteration. Note that we do not require the DPM to converge very early, since the skill posterior is also not well-trained. Additionally, we compute the soft KL divergence as a weighted sum between the skill posterior distribution and the DPM

components: $\sum_{k=1}^{K} p_{ik} \mathbb{KL}(q(z|a_i)||\mathcal{N}(\mu_k, \Sigma_k))$, where $K$ is is the current number of components, $p_{ik}$ denotes the probability of assigning the $i$-th data point to the $k$-th component, and $\mu_k, \Sigma_k$ represent the parameters of the $k$-th Gaussian component. The latent embeddings contain the respective non-parametric properties from the origin, which can be described by the DPM. This weighted sum helps to reduce assignment errors by aligning the embeddings with DPM components.

Next, we freeze the DPM parameters and update the networks. A batch of temporally extended skills with horizon length $L$ is fed into the skill posterior encoder $q(z|a_i)$, generating embeddings in the latent space. Subsequently, the skill decoder $p(a_i|z, s_t)$ receives both embeddings and the initial state $s_t$ of the skill at time t, forming a closed loop that generates estimated actions. By comparing these estimated actions with the ground truth, we calculate the reconstruction error, which is used to update the network parameters. To guide downstream learning, we simultaneously learn a skill prior $p(z|s_t)$ over the embeddings. We calculate the KL divergence between the skill posterior and the predicted skill prior $\mathbb{KL}(q(z|a_i)||p_\vartheta(z|s_t))$ to ensure the prior captures the non-parametric nature embedded in the representations. Here, we use reverse KL divergence $\mathbb{KL}(q, p)$ to enforce mode-covering.

The overall objective in this phase is to capture the non-parametric nature of skills, allowing the model to represent an unknown number of skill features that can adapt based on observed data. The total skill prior

learning objective includes three components:

$$\mathcal{L}_{total} = \zeta_1 \mathcal{L}_{rec}(\boldsymbol{a}_i, \hat{\boldsymbol{a}}_i)$$
$$+ \zeta_2 \sum_{k=1}^{K} p_{ik} \mathcal{L}_{KLD}(q(z|\boldsymbol{a}_i), \mathcal{N}(\mu_k, \Sigma_k))$$
$$+ \zeta_3 \mathcal{L}_{KLD}(p_{\vartheta}(z|s_t), q(z|\boldsymbol{a}_i)), \quad (7)$$

where $\{\zeta_i\}_{i=1,2,3}$ are the respective weighting factors, $\mathcal{L}_{rec}(\boldsymbol{a}_i, \hat{\boldsymbol{a}}_i)$ represents the mean squared reconstruction error between generated actions $\hat{a}_i$ and ground truth $a_i$, $\mathcal{L}_{KLD}$ is the KL divergence between two distributions. This pretraining phase aims to build a flexible skill prior that can represent diverse actions, providing a foundation for the downstream RL policy to focus on meaningful action patterns without having to learn from scratch.

### C. Downstream Long-Horizon RL with Skill Priors

In the downstream task learning phase, as illustrated on the right side of Fig.1 (highlighted in yellow), we use a hierarchical policy structure. The upstream inference module, implemented with the state-of-the-art (SOTA) Soft Actor-Critic (SAC) framework, learns to produce latent skill embeddings $z$ based on observations and the pre-trained non-parametric skill prior. For the downstream component, we integrate the pre-trained skill decoder $p(\boldsymbol{a}_i|z, s_t)$ from Phase I, with its parameters kept fixed. The decoder takes in the latent embeddings and outputs a sequence of actions $\{a_t, a_{t+1}, \cdots, a_{t+L}\} \sim p(\boldsymbol{a}_i|z, s_t)$ of length $L$, continuing until the next state update. Thus, downstream task learning can be formulated as a standard MDP problem with temporal abstraction. Following Sutton's temporal abstraction framework [18], we replace single-step rewards $r$ with $L$-step cumulative rewards $\tilde{r} = \sum_{t=1}^{L} r_t$, single-step state transition with $L$-step transitions $s_{t+L} \sim p_{\varphi}(s_{t+H}|s_t, z_t)$.

When learning from a large dataset containing a diverse set of primitive skills, the policy may face exploration challenges due to the rich skill space. To address this, we use the pre-trained Bayesian non-parametric skill prior to guide the upstream inference module, improving exploration efficiency. This skill prior can be naturally incorporated into a maximum-entropy RL framework. Given Eq.(6), the policy entropy term is effectively equivalent to the negative KL divergence between the policy distribution and our pre-trained skill prior, up to a constant. To guide the upstream inference module's exploration in skill space, we replace the traditional entropy term with this KL divergence, which encourages alignment with the skill prior distribution.

The objective for the upstream module is defined as:

$$J(\varphi) = \mathbb{E}_\pi \left[ \sum_{t=1}^{T} \tilde{r}(s_t, z_t) - \omega \mathbb{KL}(\pi_\varphi(z_t|s_t), p(z_t|s_t)) \right] \text{ with}$$
$$H(\pi_\varphi(\cdot|s_t)) = -\mathbb{E}_\pi[log\pi(\cdot|s_t)] \propto -\mathbb{KL}(\pi_\varphi(\cdot|s_t), p(\cdot|s_t)), \quad (8)$$

where $\varphi$ represents the parameters of the upstream module. Additionally, Fig. 2 illustrates our proposed skill distributions to enable intuitive understanding.
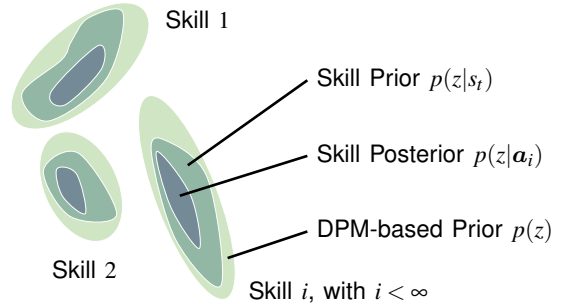


Fig. 2. Skill distribution assumption of our proposed framework.

## V. EXPERIMENTS

In this section, we present the experimental performance of our framework. We begin with a description of our experiment setup, followed by an analysis of the training performance of our Bayesian non-parametric skill prior in Phase I. Finally, we evaluate our agent's performance in downstream long-horizon RL tasks. Compared to state-of-the-art baselines, our framework shows substantial improvements in accelerating downstream task exploration.

### A. Experiment Setup

We conduct our experiments in the Franka Kitchen environment [5], a simulated scenario where a 9-DoF Franka robot arm manipulates various kitchen appliances and objects, including a microwave, burners, cabinets, etc. The scene consists of seven subtasks in total. We use a dataset from the D4RL benchmark [19], which provides 400 teleoperated, unstructured sequences demonstrating diverse manipulation tasks. For skill prior training, we use the "mixed" version of the dataset, which includes demonstrations for four subtasks (microwave, kettle, bottom burner, and light switch). These tasks are presented as unstructured skills in action slices, leveraging solely these skills can not accomplish the total long-horizon task. In the downstream long-horizon learning phase, the agent must learn to select and combine skills to complete the long-horizon tasks with four sub-goals in an unseen order. The training objective is to maximize the total sparse episode reward, where agents
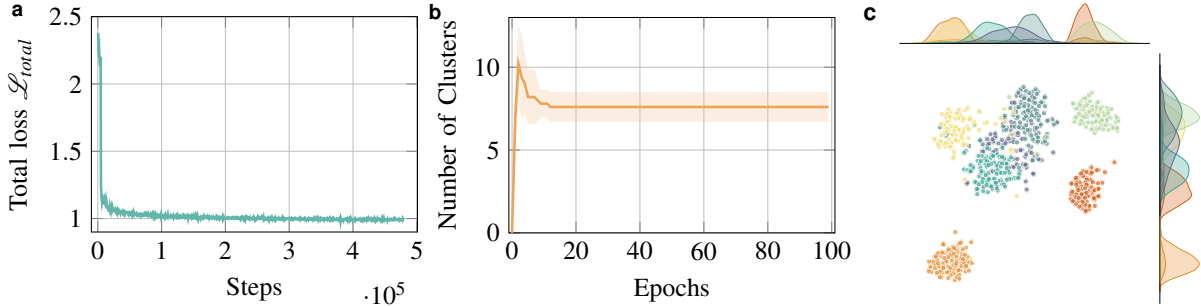
Fig. 3. The evaluation of Bayesian non-parametric skill prior. **a**, The total training loss ($\mathscr{L}_{total}$) observed during the skill prior pretraining phase. **b**, The evolution of the number of generated clusters in the Bayesian non-parametric skill prior space throughout training. We conduct at least five trials and report the mean and standard deviation ($\mu \pm \sigma$). **c**, t-SNE projection of the DPM-based skill prior in the final epoch.

award a score of 1 only for completed subtasks and 0 otherwise. Additionally, three extra subtasks (top burner, slide cabinet, and hinge cabinet) are introduced into the task sequence to assess the agent's zero-shot adaptation performance. To ensure reliable results, we conduct trials with at least five random seeds and report the mean and standard deviation for quantitative comparisons ($\mu \pm \sigma$).

### B. Representation Learning of Skill Prior

We train the proposed Bayesian non-parametric skill prior model as presented in Sec. IV-B. To evaluate the training performance, we first examine the total regression loss during training, as depicted in Fig. 3a. The training loss converges rapidly, stabilizing around a value of 1.0. In Fig. 3b, we illustrate the evolution of the number of components in the Bayesian non-parametric skill prior. Initially, the DPM starts with a single Gaussian component. After each training epoch, the DPM uses buffered data collected during the epoch to adjust the number, shape, and density of components, adapting to the data representation. In the early stages, when both the network parameters and the DPM still do not converge, the DPM may generate extra components, in some cases reaching up to 12. As training progresses and data noise decreases, the DPM merges redundant components to optimize its objective, ultimately stabilizing around 6 - 8 components in the skill prior space. Fig. 3c displays the t-SNE [20] projection of the Bayesian non-parametric skill prior space after training. The results indicate that our DPM-based prior model effectively identifies and clusters the underlying features in the data, providing a well-structured skill prior space that can guide the decoder's action pattern learning and enhance subsequent downstream RL training for long-horizon manipulation.

### C. RL for Long Horizon Manipulation

In the subsequent experiments, we compare the downstream long-horizon task performance of our framework,

HELIOS, to recent SOTA baseline models in our proposed RL manipulation tasks, as outlined below:

- **Standard SAC**: We train the agent from scratch using the SAC [11]. This baseline assesses the benefits of leveraging prior experience.
- **Behavioral Cloning (BC)**: We use a BC policy trained on the given dataset, then finetune it in the downstream tasks.
- **Skill-Space Policy (SSP)**: This variation trains a high-level policy in the skill embedding space without using a skill prior, examining the role of a learned skill prior in enhancing downstream task learning.
- **Flat Prior**: This baseline learns a single-step action prior without GRU/LSTM modules in the backbone, which regularizes downstream learning as a form of temporal abstraction.
- **Skill-Prior RL (SPIRL)**: This framework [1] incorporates a skill prior within a hierarchical RL framework, similar to our approach, but models the skill prior as a single fixed Gaussian distribution.
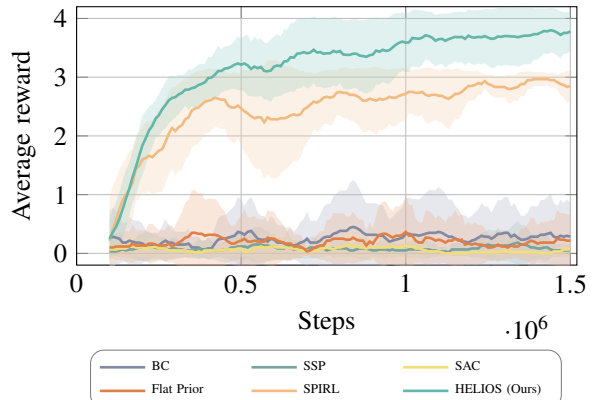


Fig. 4. Average reward of total long-horizon manipulation task. In the Franka-Kitchen Benchmark, we use sparse rewards to train the agent, awarding a score of 1 only for successfully completed subtasks and 0 otherwise. For each model, we run at least five trials, reporting the average reward and standard deviation ($\mu \pm \sigma$) for comparison.
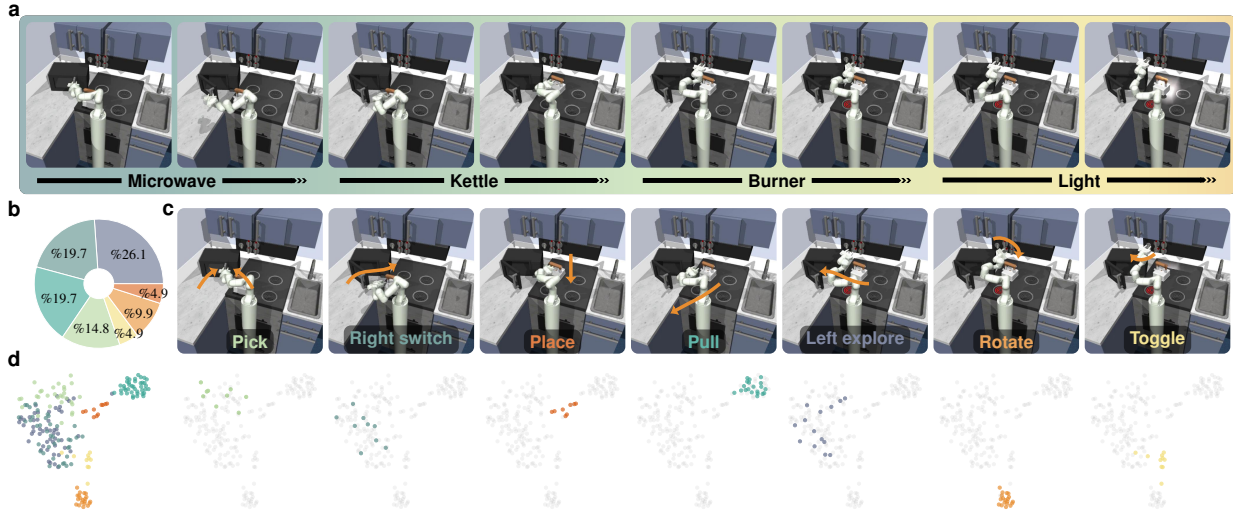
Fig. 5. Long-horizon manipulation task performance. **a**, Snapshots of each sub-task, demonstrating that the agent efficiently completes all assigned subtasks. **b**, The average skill ratios applied across the entire task, based on the results from at least five trials. **c**, Snapshots of each primitive skill motion. Our Bayesian non-parametric prior captures seven base skills: "pick", "place", "pull", "rotate", "toggle", and "explore" movements (both left and right directions). **d**, t-SNE projections of each primitive skill motion, showing the re-encoded skills through the pre-trained skill encoder $q(z|a_i)$ and their corresponding assignments in the Bayesian non-parametric knowledge space.

This SOTA framework assesses the impact of using a simpler, fixed skill prior as opposed to our adaptive, non-parametric approach.

*a) Quantitative statistics:* As shown in Fig. 4, our proposed framework HELIOS substantially outperforms all baseline models in terms of average reward on long-horizon manipulation tasks. HELIOS quickly reaches a high average reward, stabilizing above 3.7 (with maximal 4.0), while other baselines either converge at lower reward values or exhibit slower progress. Notably, SPIRL performs better than some baselines due to its use of a skill prior, but it falls short compared to HELIOS, which leverages a more flexible Bayesian non-parametric skill prior modeled with DPM and Mem-oVB heuristics. This adaptive skill prior allows HE-LIOS to dynamically capture an optimal set of skills that are more expressive and well-suited for complex, long-horizon tasks. Although Flat Prior incorporates a trained skill prior module with a simple fully connected backbone, this design fails to effectively capture the underlying skill structure and context, leading to poor performance in downstream tasks. This highlights the contribution of our proposed GRU-based skill prior network. Without the guidance of skill prior, the SAC, BC, and SSP models struggle in this complex, sparse-reward, long-horizon task environment, achieving an average reward of less than 0.5. By utilizing the DPM-based skill prior, HELIOS benefits from efficient exploration and the ability to recombine learned skills in new ways, essential for tackling unseen task sequences in long-horizon scenarios. This structured and adaptable skill

space enables faster and more stable convergence, highlighting HELIOS's superior performance in the challenging Franka Kitchen environment.

*b) Visulization:* Figure 5a demonstrates the rendering snapshots of our proposed agent performing a sequence of subtasks in the Franka Kitchen environment, including manipulating the microwave, kettle, burner, and light switch. The images demonstrate that the agent completes all assigned subtasks in the scene, highlighting its ability to handle complex, multi-step, long-horizon tasks. The renderings of these subtasks reflect the agent's capability to utilize the pre-trained Bayesian non-parametric skill prior for efficient skill selection and execution. This showcases HELIOS's strength in combining learned skills to solve sequential subtasks in the long-horizon manipulation scene.

The pie chart in Figure 5b illustrates the distribution of skill ratios utilized throughout the long-horizon task sequence, averaged over at least five trials and color-coded by respective skills. In our task dataset, the Bayesian non-parametric skill prior captures an average of seven primitive skill motions, with snapshots of each skill depicted in Figure 5c. These skills include "pick" (14.8%), "place" (4.8%), "pull" (19.7%), "rotate" (9.9%), "toggle" (4.9%), and "explore" in both left (26.1%) and right (19.7%) directions. Together, these skills form the fundamental building blocks of the agent's actions, enabling efficient recombination to perform long-horizon tasks. By leveraging the pre-trained Bayesian non-parametric skill prior, the agent effectively utilizes these skills to manage complex ob-

ject manipulations, showcasing the prior's capability to encode and generalize key motion behaviors. Additionally, Figure 5d presents the t-SNE projections of each primitive skill motion in the latent space, revealing well-defined clusters of re-encoded skills generated by the skill encoder $q(z|a_i)$ from phase I. Each cluster corresponds to a distinct skill, highlighting the ability of the Bayesian non-parametric prior to group similar actions while maintaining clear boundaries between different skill types. This structured latent space demonstrates the prior's strength in capturing complex, multi-modal skill distributions. Such precise clustering is essential for guiding the agent in downstream RL tasks, enabling efficient exploration and robust action pattern inference.

*c) Skill adaptation:* Furthermore, we evaluate the zero-shot skill adaptation capability of our proposed framework in unseen task scenarios. To do this, we randomly incorporate the subtasks "open slider cabinet", "open hinge cabinet", and "turn on top burner" into the long-horizon task sequence. Importantly, the skills required to accomplish these subtasks are not present in the original data library. An example of the task rendering is shown in Figure 6. Our agent successfully adapts the knowledge and skills learned from other tasks to handle these unseen scenarios, ultimately completing the new subtasks. This highlights the generalization capability of our framework, showcasing its ability to apply prior knowledge to previously unencountered tasks.
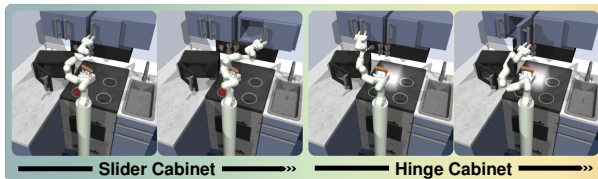


Fig. 6. Zero-shot skill adaptation for unseen sub-tasks. Videos are available at `https://ghiara.github.io/HELIOS/`.

## VI. CONCLUSIONS

In this work, we proposed HELIOS, a framework that integrates a Bayesian non-parametric skill prior with hierarchical RL to address long-horizon manipulation tasks. Experiments in the Franka Kitchen environment demonstrated its superior performance and generalization ability compared to state-of-the-art baselines. In future work, we aim to extend HELIOS by combining it with foundation models to enhance scalability and explore its application in real-world long-horizon tasks.

## ACKNOWLEDGMENT

## REFERENCES

[1] K. Pertsch *et al.*, "Accelerating reinforcement learning with learned skill priors," in *Conference on robot learning*. PMLR, 2021, pp. 188–204.

[2] T. Yu *et al.*, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on robot learning*. PMLR, 2020, pp. 1094–1100.

[3] M. C. Hughes and E. Sudderth, "Memoized online variational inference for dirichlet process mixture models," *Advances in neural information processing systems*, vol. 26, 2013.

[4] K. Cho, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[5] A. Gupta *et al.*, "Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning," in *Conference on Robot Learning*. PMLR, 2020, pp. 1025–1037.

[6] C. Finn *et al.*, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.

[7] K. Rakelly *et al.*, "Efficient off-policy meta-reinforcement learning via probabilistic context variables," in *International conference on machine learning*. PMLR, 2019, pp. 5331–5340.

[8] Z. Bing, A. Koch, X. Yao, K. Huang, and A. Knoll, "Meta-reinforcement learning via language instructions," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5985–5991.

[9] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration," in *International conference on machine learning*. PMLR, 2019, pp. 2052–2062.

[10] A. Kumar *et al.*, "Stabilizing off-policy q-learning via bootstrapping error reduction," *Advances in neural information processing systems*, vol. 32, 2019.

[11] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[12] Z. Bing, L. Knak, L. Cheng, F. O. Morin, K. Huang, and A. Knoll, "Meta-reinforcement learning in nonstationary and nonparametric environments," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2023.

[13] N. Y. Siegel *et al.*, "Keep doing what worked: Behavioral modelling priors for offline reinforcement learning," *arXiv preprint arXiv:2002.08396*, 2020.

[14] D. A. Reynolds *et al.*, "Gaussian mixture models." *Encyclopedia of biometrics*, vol. 741, no. 659-663, 2009.

[15] E. Nalisnick and P. Smyth, "Stick-breaking variational autoencoders," *arXiv preprint arXiv:1605.06197*, 2016.

[16] P. Goyal *et al.*, "Nonparametric variational auto-encoders for hierarchical representation learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5094–5102.

[17] D. M. Blei and M. I. Jordan, "Variational inference for dirichlet process mixtures," *Bayesian Analysis*, vol. 1, no. 1, pp. 121–144, 2006.

[18] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial intelligence*, pp. 181–211, 1999.

[19] J. Fu *et al.*, "D4rl: Datasets for deep data-driven reinforcement learning," *arXiv preprint arXiv:2004.07219*, 2020.

[20] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, 2008.