Learning and Retrieval from Prior Data for Skill-based Imitation Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract: Imitation learning offers a promising path for robots to learn general-purpose tasks, but traditionally has enjoyed limited scalability due to high data supervision requirements and brittle generalization. Inspired by recent work on skill-based imitation learning, we investigate whether leveraging prior data from previous related tasks can enable learning novel tasks in a more robust, data-efficient manner. To make effective use of the prior data, the agent must internalize knowledge from the prior data and contextualize this knowledge in novel tasks. To that end we propose a skill-based imitation learning framework that extracts temporally-extended sensorimotor skills from prior data and subsequently learns a policy for the target task with respect to these learned skills. We find a number of modeling choices significantly improve performance on novel tasks, namely representation learning objectives to enable more predictable and consistent skill representations and a retrieval-based data augmentation procedure to increase the scope of supervision for the policy. On a number of multi-task manipulation domains, we demonstrate that our method significantly outperforms existing imitation learning and offline reinforcement learning approaches.

Keywords: Imitation Learning, Skill Learning, Robot Manipulation

1 Introduction

A long-standing dream of robotics researchers is to enable robots to perform general-purpose tasks in diverse environments. In recent years, imitation learning has led great progress towards this goal. Recent work has demonstrated that well-engineered behavioral cloning methods can achieve competitive performance on diverse manipulation tasks [1, 2, 3]. Despite this promise, these imitation learning algorithms have been confined to relatively small-scale domains. For real-world problems, current imitation learning algorithms often demand a large number of task demonstrations, placing a considerable burden on human end users demonstrating tasks.

Realizing these limitations, a series of recent work seek to use prior interaction data to improve the sample efficiency of imitation learning on new tasks. Such prior data can come in various forms, including task-agnostic exploratory "play" data [4] or demonstrations previously collected for different tasks [5]. An open question is how best to extract knowledge from these large prior datasets and use this knowledge to facilitate learning novel tasks. One promising approach is skill-based imitation learning [6, 7, 8], which aims to learn a latent space of short-horizon sub-trajectories from the prior data (called *skill learning*) and subsequently learn a policy to invoke the skills to solve a specific downstream task (called *policy learning*). This approach offers several appealing advantages: First, the policy benefits from the temporal abstraction encapsulated by the skills, allowing the policy to focus on higher-level reasoning about *what* behavior to perform rather than *how* to execute that behavior; and second, by reasoning about the target task in relation to skills trained on prior data, the robot implicitly distills knowledge from the rich diverse interactions of the prior data into the policy. Even so, existing skill-based imitation learning approaches bring forth modest im-

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

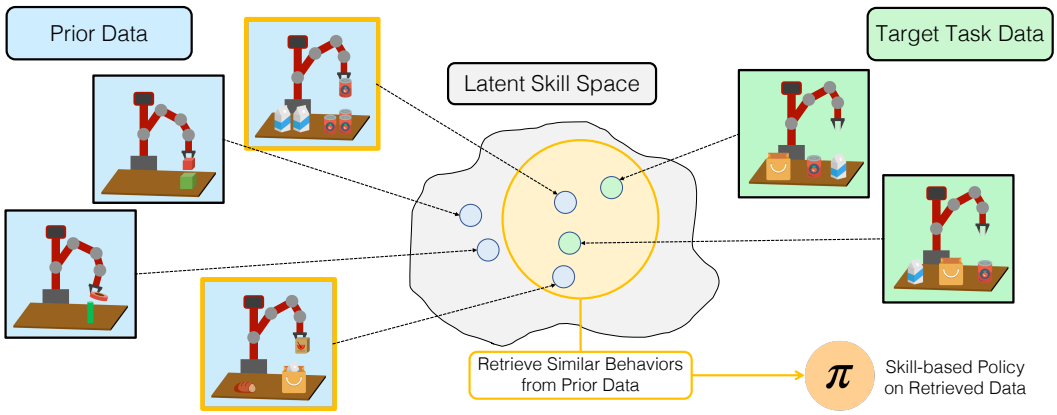

Figure 1: **Overview.** We present a skill-based imitation learning framework that learns a consistent latent representation of skills, with similar behaviors mapped to similar regions of the latent space. Given a dataset of target task demonstrations we use this latent space to retrieve similar behaviors from the prior data. We then train a skill-based policy on the target task data and retrieved behaviors from the prior data.

provements over simple behavioral cloning methods. This work aims at identifying the underlying limitations of existing approaches and designing new methods to address these limitations.

We argue that a practical approach should encompass two key properties. First, the latent skill pace should serve as a *consistent* representation for downstream policy learning, allowing the policy to infer which appropriate skill to use in new situations accurately. At a high-level, with a consistent representation behaviors that are semantically similar are mapped to similar regions of the latent space. While prior methods [9, 7, 6, 8] have widely adopted variational autoencoders to transform sub-trajectories into embeddings, we empirically show that this learning objective alone is insufficient for constructing a consistent latent space. As a result, it increases the likelihood of the policy executing irrelevant skills. Second, the agent should take advantage of the prior data to learn the skills and the policy. A major limitation of existing approaches is that they only focus on using the prior data for skill learning but not policy learning. Consequently, policies learned on a small number of demonstrations on the target task are prone to severe overfitting and covariate shift.

To address these limitations, we introduce **S**kill-**A**ugmented **I**mitation **L**earning with pri**or** **R**etrieval (SAILOR). To improve the predictability of the skill representation, SAILOR uses an auxiliary temporal consistency objective that predicts the temporal distance between sub-trajectories in the demonstration sequences. To use prior data for policy learning, we introduce a novel retrieval-based data augmentation procedure that selectively retrieves data relevant to the target task. Specifically, we consider a subset of sub-trajectories from the prior dataset that have high latent skill similarity with sub-trajectories in the target task demonstrations. We evaluate on a wide variety of manipulation tasks in simulation and the real world, where SAILOR significantly outperforms the state-of-the-art imitation learning and offline reinforcement learning approaches. Through comprehensive analysis, we highlight the roles that prior data, our auxiliary representation learning objective, and retrieval-based data augmentation have in data-efficient learning of robust manipulation policies.

2 Related Work

Learning from Human Demonstrations. Recent work has shown promise for learning manipulation tasks using human demonstrations. These works span methods in demonstration-guided reinforcement learning [10, 7, 11, 12, 13, 14], behavioral cloning [1, 15, 15, 16, 17], and offline reinforcement learning [18, 19]. While these methods show promising results, they typically require large amounts of supervision and exhibit brittle generalization on more complex, long-horizon tasks. A number of lines of work have sought to address these limitations. Some have investigated using inductive biases to better generalize to novel settings [2, 3, 20] while others have investigated

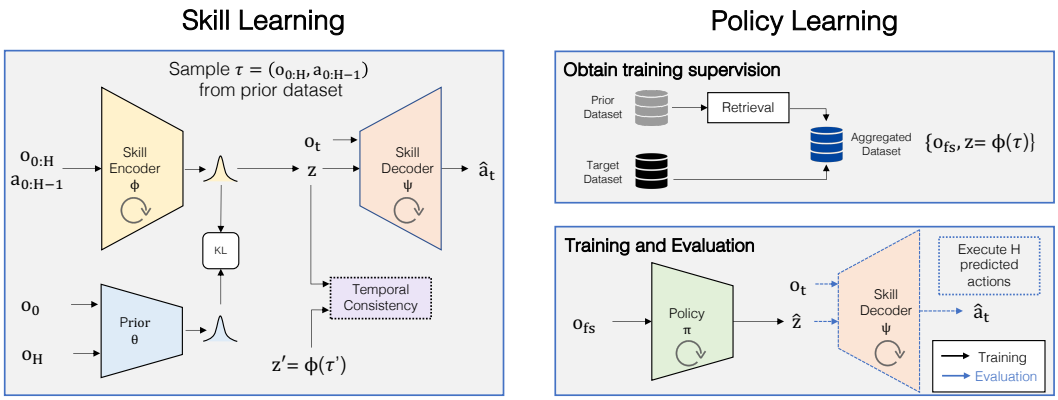

Figure 2: **Model Overview.** Our method consists of a skill learning and policy learning phase. (left) In the skill learning phase we learn a latent skill representation of sub-trajectories via a variational autoencoder, and we include an additional temporal consistency term to learn a more consistent latent representation. In the policy learning phase we first obtain training supervision for the policy by aggregating sub-trajectories in the target dataset in addition to retrieving sub-trajectories in the prior dataset that similar to those in the target dataset. Given this aggregated data, given a sampled sub-trajectory we then train the policy to predict the latent skill given a history of observations preceding the sub-trajectory.

incorporating human feedback to more effectively recover from agent failures [21, 22]. While these efforts have shown some improvements, they also introduce new assumptions that limit their scope to a specific types of tasks, or introduce additional computational complexities and new sources of human labor that limit their practicality.

Accelerating Robot Learning with Prior Data. Alternatively, some works have leveraged large offline prior datasets to accelerate learning on downstream robotic tasks. These prior datasets can take on various forms, such as task-agnostic play data [4, 23], demonstrations previously collected for different tasks [5], self-supervised agent-generated data [24], or more unconstrained data sources such as video clips depicting human behaviors [25, 26, 27]. Each of these different types of data affords certain advantages and disadvantages. On one hand, large vision datasets depicting human behaviors are readily available and recent work has leveraged these datasets to learn pre-trained visual representations for downstream control tasks [28, 29]. Yet despite their appeal, after experimenting with one such approach, R3M [28], we found that it can sometimes hinder downstream performance. One hypothesis is that the prior data on which these methods are trained on exhibit significant domain shift compared to downstream robot tasks, making it challenging to extract useful representations. In this work we instead consider large multi-task datasets.

Multi-Task Imitation Learning. Within the context of learning from large prior data, there is a large body of work on multi-task imitation learning. These includes task-conditioned [5], language-conditioned [30, 4, 31, 32], and skill-based [6, 7, 8] imitation learning. While most task-conditioned and language-conditioned multi-task imitation approaches seek to learn a single policy that performs well across a series of tasks, we focus on learning task-specific policies by utilizing a large multi-task prior dataset to learn a useful skill representation space, and supervising the task-specific policy using learned latent skills. These multi-task imitation approaches could be complimentary to our skill representation learning pipeline, however, we use a VAE-based approach to learn a latent skill space as in prior skill-based imitation works. In contrast to other skill-based imitation learning (see Sec. 3) approaches, we enforce two key properties – a temporal consistency objective in our learned skill representation, and a retrieval-based mechanism to improve task-specific policy learning.

3 Problem Formulation

Our overall goal is to leverage prior data to effectively learn novel target tasks in a data-efficient manner. Formally, we consider a target task as a Markov Decision Process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, p, p_0, \gamma)$ representing the state space, action space, reward function, transition probability, initial state distri-

bution, and discount factor. We assume the agent has access to observations in $\mathcal{O}$ and our objective is to maximize the discounted sum of rewards for the task. To learn the policy, we assume access to a small set of demonstrations $\mathcal{D}_{\text{target}}$ collected for the target task and a large set of demonstrations $\mathcal{D}_{\text{prior}}$ either from previous related tasks or task-agnostic interactions. Each demonstration is a variable-length trajectory $\{o_0, a_0, o_1, \cdots, o_{T_i}\}$ with $o_i \in \mathcal{O}$ denoting the agent's observation and $a_i \in \mathcal{A}$ denoting the agent's action. We assume that all demonstrations across $\mathcal{D}_{\text{prior}}$ and $\mathcal{D}_{\text{target}}$ share a common observation and action space. Given these two datasets, our goal is to learn a policy π that maps observations to actions. We highlight that $\mathcal{D}_{\text{prior}}$ and $\mathcal{D}_{\text{target}}$ may have significant differences, i.e. the two datasets may come from different environments, be collected by different human demonstrators, and demonstrate different tasks.

Skill-based Imitation Learning. We adopt skill-based imitation learning [7, 6, 8] as the underlying framework for our approach. This framework consists of two stages, skill learning and policy learning. In the skill learning phase, we learn a skill embedding space $\mathcal{Z} \subset \mathbb{R}^d$ of short-horizon sub-trajectories $\tau = \{o_o, a_0, o_1, \cdots, o_{H-1}, a_{H-1}, o_H\}$ in $\mathcal{D}_{\text{prior}}$. These skills serve as an abstract representation of the agent's behavior and can be composed to solve a range of downstream tasks. A number of representation learning methods can be employed to learn the skill embeddings, spanning reconstruction-based methods and contrastive learning. In the subsequent policy learning phase, we are given $\mathcal{D}_{\text{target}}$ and our goal is learn a policy π for the target task $\mathcal{T}$. Instead of emitting actions $a \in \mathcal{A}$, the policy now emits *skills* $z \in \mathcal{Z}$ and during execution we decode z into a sequence of actions $\{a_0, a_1, \cdots, a_{H-1}\}$ via a skill decoder model p_ψ. To learn the policy, prior work has proposed parametric [6] and non-parametric [8] polices that first parse segments of $\mathcal{D}_{\text{target}}$ into skills and subsequently learn to map observations in $\mathcal{D}_{\text{target}}$ to these skills.

4 Skill-based Imitation Learning with Retrieval

In this section, we describe our *skill-based imitation learning* approach that can leverage prior multi-task data to efficiently learn novel target tasks with a small amount of task-specific demonstrations. As discussed in Sec. 3, this consists of two phases – a task-agnostic skill learning phase, where a latent skill space is learned using the prior data, and a task-specific policy learning phase where task-specific data is used to learn a policy using the skills as supervision. Compared to prior methods, our approach makes two key considerations – we (1) ensure that the learned skill space is a *consistent* representation for downstream policy learning, and (2) improve the efficacy of task-specific policy learning by *retrieving* task-relevant datapoints from the prior dataset.

4.1 Learning a Consistent Representation of Skills

We learn a skill representation by encoding sub-trajectories with a variational autoencoder (VAE) and we introduce an auxiliary term to the objective function to shape the representation. Denoting a given sub-trajectory as τ, we employ an LSTM encoder q_ϕ that encodes τ into a Gaussian distribution over latent skills. Our decoder is an LSTM network p_ψ that, for each timestep t, decodes a latent z and the given observation o_t into the reconstructed action $\hat{a}_t$. We additionally employ a learned prior p_θ to encourage sub-trajectories with similar starting and ending observations to have similar latent representations [23]. Our VAE loss objective is then

$$\mathcal{L}_{\text{VAE}}(\phi, \psi, \theta) = -\mathbb{E}_{z \sim q_\phi(z|\tau)} \left[\sum_{t=0}^{H-1} \log p_\psi(a_t | z, o_t) \right] + D_{KL}(q_\phi(z|\tau) || p_\theta(z|o_0, o_H)) \quad (1)$$

However, action reconstruction is not the sole objective of our skill learning model – learning consistent and predictable representations of behavior is critical for downstream policy learning, as shown by recent work [33, 34]. While the KL divergence term is one step towards this objective (by encouraging skills to be predictable given partial information from the sub-trajectories), in this work we introduce an additional *temporal consistency* term that encourages the learned latent space to predict the temporal difference between two sub-trajectories. Specifically, given two-sub-trajectories τ_1 and

τ_2 from the same trajectory separated by t timesteps, we learn a model m_ω to predict t given the corresponding skill mean embeddings of the trajectories:

$$\mathcal{L}_{\text{TC}}(\omega, \phi) = \Big(m_\omega\Big(\mu(q_\phi(z|\tau_1)), \mu(q_\phi(z|\tau_2))\Big) - t\Big)^2 \tag{2}$$

We back-propagate $\mathcal{L}_{\text{TC}}$ through the skill encoder model, allowing the this term to shape the learned skill representation. Our overall objective is a weighted combination of the VAE and temporal consistency objectives:

$$\mathcal{L}_{\text{Skill}}(\phi, \psi, \theta, \omega) = \mathcal{L}_{\text{VAE}}(\phi, \psi, \theta) + \alpha\mathcal{L}_{\text{TC}}(\omega, \phi) \tag{3}$$

4.2 Retrieval-based Policy Learning

In the policy learning phase we employ an LSTM policy that maps a frame history of observations to the skill z to execute next. To obtain supervision for the policy, we assume that we are given a dataset of sub-trajectories $\mathcal{D}_{\text{policy}} = \{(o^i_{fs}, \tau_i)\}$, where o^i_{fs} corresponds to the frame-stacked history of F observations preceding the H-length sub-trajectory τ_i. We encode τ_i via the skill model encoder q_ϕ into its mean latent skill embedding $z = \mu(q_\phi(\tau))$ and train the policy to predict z from o^i_{fs}. During execution, we roll out the skill decoder p_ψ to obtain predicted actions $a_t = \mu(p_\psi(.|z, o_t))$. After rolling out the skill for H timesteps we repeat the process by sampling a new skill from the policy.

A common approach for skill-based policy learning is to train the policy by using all H-length sub-trajectories in $\mathcal{D}_{\text{target}}$, but this limits the amount of supervision, especially if $\mathcal{D}_{\text{target}}$is small. On the other hand, naively training on all sub-trajectories in $\mathcal{D}_{\text{target}}$ and $\mathcal{D}_{\text{prior}}$ can hurt performance [35]. We instead propose a retrieval-based mechanism to only train on sub-trajectories in $\mathcal{D}_{\text{prior}}$ that have high similarity with those in $\mathcal{D}_{\text{target}}$. While many similarity metrics are suitable, in this work we measure similarity with respect to the skill embedding space – intuitively, sub-trajectories with similar skill embeddings demonstrate similar behaviors. First, we obtain skill embeddings of randomly sampled sub-trajectories in $\mathcal{D}_{\text{prior}}$ and $\mathcal{D}_{\text{target}}$:

$$Z_{\text{prior}} = \{\mu(q_\phi(\tau_i))\}, \tau_i \sim \mathcal{D}_{\text{prior}}; \qquad Z_{\text{target}} = \{\mu(q_\phi(\tau_j))\}, \tau_j \sim \mathcal{D}_{\text{target}} \tag{4}$$

We then calculate the pairwise L_2 distances D between the prior and target dataset skill embeddings, i.e. $D_{ij} = \|Z^i_{\text{prior}} - Z^j_{\text{target}}\|_2$. Next, for each prior dataset skill $z_i \in Z_{\text{prior}}$ we find the closest corresponding target dataset skill: $M_i = \min_j D_{ij}$ Finally we retrieve the top-n sub-trajectories in $\mathcal{D}_{\text{prior}}$ with the smallest distance `argsort(M)[:n]`. We then train the policy the aggregated set of retrieved sub-trajectories from $\mathcal{D}_{\text{prior}}$ and all sub-trajectories in $\mathcal{D}_{\text{target}}$. We additionally modify the policy with an input binary bit i indicating whether the input corresponds to the target dataset or retrieved dataset.

5 Experiments

5.1 Simulated Experiment Setup

We perform empirical evaluations on two simulated robot manipulation domains from prior work:

Franka Kitchen [11]: A simulated kitchen environment involving different sub-tasks, such as opening cabinets, moving a kettle, opening a microwave, and turning on a stove. This environment comes with a large dataset of approximately 600 demonstrations performing various permutations of seven subtasks. In this dataset, a subset of 18 demonstrations correspond to $\mathcal{D}_{\text{target}}$ and demonstrate a specific permutation of subtasks: opening the microwave, followed by moving the kettle, flipping on the light switch, and opening the sliding cabinet. We consider two prior datasets $\mathcal{D}_{\text{prior}}$: (1) using all demonstrations except the ones corresponding to the target task (`Kitchen-All`); and (2) using all demonstrations except those that involve interacting with the microwave (`Kitchen-No Microwave`). These prior datasets have 584 and 235 demonstrations, respectively. The latter is more challenging as it does not cover all subtasks that the agent will perform for the test task.

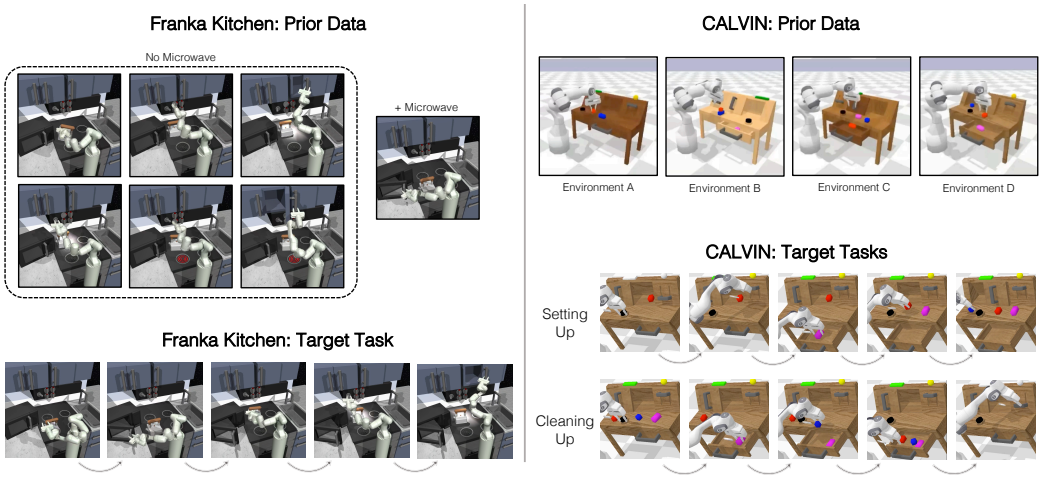

Figure 3: **Simulated Tasks.** We perform extensive evaluations on two simulation domains. (left) Franka Kitchen: our target task involves a specific permutation of four subtasks and we consider two prior datasets: demonstrations involving all subtasks, and demonstrations involving all subtasks except opening the microwave. (right) CALVIN: we adopt the play dataset of Mees et al. [4] as our prior data and perform evaluations on two target tasks: setting up the playroom environment and the converse, cleaning up the environment.

CALVIN [4]: A simulated tabletop playroom environment comprising a drawer, cubbies, two lights, and three blocks. Accompanying this environment is a large dataset of task-agnostic "play" data comprising 2.3M transitions. The play data encompass diverse behaviors, such as opening and closing drawers, turning on and off the lights, and picking, placing, and pushing blocks. We use all play data as $\mathcal{D}_{\text{prior}}$ to solve two target tasks. The first target task involves setting up the playroom environment in multiple stages (`CALVIN-Setting Up`). Specifically, the robot must turn on the lights, retrieve the pink block from the drawer, place it on the table, retrieve the red and blue blocks from the cubby area, and place them on the table. The second target task in contrast involves cleaning up the playroom environment (`CALVIN-Cleaning Up`). Specifically, the robot must open the drawer, place all three blocks into the drawer, close the drawer, and turn off the lights. For each task, we collect 30 demonstrations, which amounts to about half an hour of data collection.

The CALVIN domain is significantly more challenging than the Franka Kitchen domain, as the target tasks have a longer horizon and involve a greater number of objects. Most importantly, in contrast to Franka Kitchen, the prior and target datasets in CALVIN are collected by *different human demonstrators* who exhibit different styles of teleoperation. As a result, the prior and target task datasets present a higher degree of distribution mismatch, making it more challenging to adopt the prior dataset for learning the target tasks.

We learn vision-based policies for both domains. We refer readers to the appendix for a detailed discussion of our environments and evaluation settings.

5.2 Quantitative Analysis

We evaluate our method against state-of-the-art imitation learning and offline reinforcement learning methods:

BC-RNN: behavioral cloning on $\mathcal{D}_{\text{target}}$without prior data. We adopt the LSTM-based BC-RNN implementation in robomimic [1], which has shown superior performance over other behavioral cloning approaches.

BC-RNN (FT): BC-RNN variant that leverages prior data. We first pre-train BC-RNN on $\mathcal{D}_{\text{prior}}$ and subsequently fine-tune on $\mathcal{D}_{\text{target}}$. This baseline aims to examine the power of supervised pre-training on interaction data for imitation learning.

Dataset	BC-RNN	BC-RNN (FT)	BC-RNN (R3M)	IQL	IQL (UDS)	FIST	Ours
Kitchen-All	32.3 ± 9.0	93.0 ± 2.3	45.0 ± 4.1	65.0 ± 7.1	74.0 ± 2.9	68.7 ± 2.5	$\mathbf{95.7 \pm 1.7}$
Kitchen-No Microwave		77.7 ± 7.6			76.0 ± 5.7	83.0 ± 0.8	$\mathbf{95.0 \pm 2.2}$
CALVIN-Setting Up	33.0 ± 2.2	72.8 ± 5.4	12.3 ± 3.4	7.7 ± 1.7	16.3 ± 2.6	3.7 ± 2.4	$\mathbf{77.3 \pm 3.1}$
CALVIN-Cleaning Up	41.0 ± 1.6	61.8 ± 9.1	13.7 ± 4.1	14.7 ± 2.1	20.0 ± 3.7	9.3 ± 5.0	$\mathbf{88.0 \pm 5.1}$

Table 1: **Quantitative evaluation on two simulation domains.** We evaluate our method against a set of six baselines and report the mean task success rate and standard deviation over three seeds (exception: six seeds for BC-RNN (FT) due to high variance). We see that our method significantly outperforms the baselines on all tasks, outperforms the most competitive baseline BC-RNN (FT) by 12.7%.

BC-RNN (R3M): behavioral cloning on $\mathcal{D}_{\text{target}}$ using a frozen R3M visual representation [28] pre-trained on the large-scale Ego4D video dataset [27]. It intends to examine the effectiveness of using visual representations trained on natural images and videos.

IQL: Implicit Q-Learning [36], a recent offline reinforcement learning method with state-of-the-art performance on the D4RL dataset [37]; trained on $\mathcal{D}_{\text{target}}$.

IQL (UDS): Implicit Q-Learning with Unlabeled Data Sharing [38], which is a variant of IQL trained jointly on $\mathcal{D}_{\text{prior}}$ and $\mathcal{D}_{\text{target}}$, where the transitions in $\mathcal{D}_{\text{prior}}$ are labeled with the minimum reward (0 for our tasks). Yu et al. [38] show that this simple data augmentation procedure can effectively leverage prior data without additional rewards annotation.

FIST: Few-shot Imitation with Skill Transition Models [8], an analogue of our method that employs a non-parametric policy to select the skill to execute next. This baseline uses the same underlying skill model as our method but a different policy learning scheme and does not involve retrieval.

We report performance of all methods in Table 1. We find that our method significantly outperforms the baselines with an average task success rate of 89.0%. Notably it outperforms the most competitive baseline BC-RNN (FT) by 12.7%. BC-RNN performs poorly as it fails to learn an effective policy from a small number of demonstrations. In comparison, BC-RNN (R3M) shows improvements on the Franka Kitchen tasks, but performs worse on the CALVIN tasks. We hypothesize that this is due to the limited generalization ability of the pre-trained visual representations. The offline reinforcement learning baselines show more promising results on the Franka kitchen tasks but struggle on the more challenging CALVIN tasks. Finally, while FIST uses the identical underlying skill model as ours, the non-parametric policy struggles to find appropriate goal observations to reach.

5.3 Ablation Study

We perform an extensive ablation study to understand the effects of various modeling choices on our method. First, we study the effect of the temporal consistency term in Eq. (2) on downstream task performance by removing it from Equation (3) (`No TC`). Next we study the role of retrieval by training the policy solely on sub-trajectories in $\mathcal{D}_{\text{target}}$ (`No Retrieval`). We also study the opposite case – training the policy on all of $\mathcal{D}_{\text{prior}}$ and $\mathcal{D}_{\text{target}}$ (`All Retrieval`). Finally we study the role of prior data on our method by training the skill and policy solely on $\mathcal{D}_{\text{target}}$ (`No Prior Data`).

We present results in Table 2 for the more challenging CALVIN tasks. First, we find that the temporal consistency objective and retrieval have a significant impact on the final performance. It is worth noting that removing these components makes our model degenerate into the skill-based imitation learning setting of OPAL [6]. In fact, these two ablations perform worse than the naïve BC-RNN (FT) baseline for the CALVIN-Cleaning Up task, indicating that both an effective skill representation and the retrieval mechanism play a critical role in skill-based imitation learning. The All Retrieval ablation also performs suboptimally — qualitatively we observe that the robot often was "distracted" and performed behaviors unrelated to the target task, likely due to the multimodal distributions in the prior data. Finally, the No Prior Data ablation validates the role of prior data in learning effective skills and policy. Comparing the No Retrieval ablation and the No Prior Data ablation, we see a $10 - 15\%$ gap in performance, and this is attributed to the fact that the skills in the No Retrieval ablation are additionally trained on the prior data. Despite the loss of performance in the No Prior

Dataset	Ours	No TC	No Retrieval	All Retrieval	No Prior Data
CALVIN-Setting Up	$\mathbf{77.3 \pm 3.1}$	68.0 ± 3.7	65.0 ± 2.2	64.3 ± 10.8	50.7 ± 6.5
CALVIN-Cleaning Up	$\mathbf{88.0 \pm 5.1}$	74.7 ± 7.9	70.0 ± 0.8	65.3 ± 7.3	60.7 ± 3.1

Table 2: **Ablation Results.** We find that temporal consistency and retrieval are critical to skill-based imitation learning, as without these components the agent struggles against a naive BC-RNN (FT) baseline. In addition, we validate that prior data plays a large role in the performance of our method.

Data ablation, we still see that it outperforms the BC-RNN baseline by a significant margin. We attribute this to the temporal abstraction afforded by our skill-based learning framework.

5.4 Real World Experiments

Finally, we showcase the efficacy of our method in the real world with a kitchen environment involving eight food items, receptacles, a stove, and a serving area. We first collect a play dataset of exploratory interactions involving the food items. We consider two target tasks: (1) setting up a breakfast table by placing the bread, butter, and milk in the serving area, and (2) cooking a meal by placing the fish, sausage, and tomato into the frying pan. We collect 30 demonstrations for each task and we evaluate our method against the most competitive baseline BC-RNN (FT). We find that while on the breakfast task both methods achieve a success rate of 76.7%, on the cook task our method significantly outperforms BC-RNN (FT) with a success rate of 73.3% compared to 23.3% for BC-RNN (FT).

6 Limitations

While our method shows significant promise, it leaves limitations that we hope to address in future work. First, acquiring large amounts of multi-task prior data is difficult and costly. To amortize the high cost, large prior multi-task datasets should be useful in a diverse range of downstream tasks, rather than a handful. In this work, we evaluate our method in a limited set of target tasks and leave it for future work to scale up the variety of tasks. We also hope (and believe) that the need for large robotic datasets will be addressed in the coming years [15, 30]. Second, our method is more computationally expensive than the BC-RNN baseline [1], due to the higher number of losses and networks used. Third, our experiments focus on domains and datasets where the prior data and test tasks are reasonably close to each other (same robot hardware and similar workspace). It would be interesting to investigate methods that are tolerant to much larger domain shifts between prior and test data. Finally, our method uses imitation learning, so our agent's proficiency is limited by the performance of the demonstrator on the target tasks.

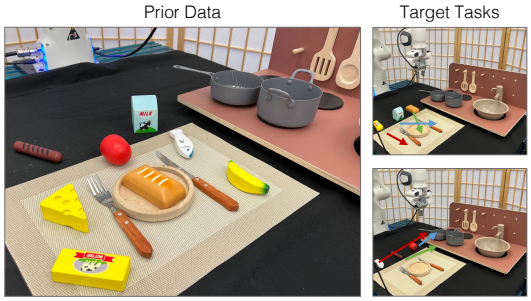

Figure 4: **Real World Tasks.** On the left, we illustrate the set of objects we use for collecting the play dataset. On the right shows the two target tasks of setting up and cleaning up the breakfast table.

7 Conclusion

We present SAILOR, a hierarchical skill-based imitation learning framework for robot manipulation. Our method uses prior data to construct a latent space of predictable and consistent skill representations. It uses these latent skills as the temporal abstraction to learn policies for vision-based manipulation. Key to its effectiveness is our newly designed representation learning objectives and retrieval-based data augmentation procedure. We demonstrate that our method can solve long-horizon manipulation tasks in simulation and on physical hardware. It brings forth a data-efficient way of programming robots with new behaviors using a small number of test task demonstrations. For future work, we plan to address the limitations we discussed in the previous section and investigate the effectiveness of this approach with various forms of prior data at different scales.

References

[1] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2021.

[2] P. Florence, C. Lynch, A. Zeng, O. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson. Implicit behavioral cloning. *Conference on Robot Learning (CoRL)*, 2021.

[3] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, and J. Lee. Transporter networks: Rearranging the visual world for robotic manipulation. *Conference on Robot Learning (CoRL)*, 2020.

[4] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard. Calvin - a benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *arXiv preprint arXiv:2112.03227*, 2021.

[5] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets, 2021. URL https://arxiv.org/abs/2109.13396.

[6] A. Ajay, A. Kumar, P. Agrawal, S. Levine, and O. Nachum. Opal: Offline primitive discovery for accelerating offline reinforcement learning. *arXiv preprint arXiv:2010.13611*, 2020.

[7] K. Pertsch, Y. Lee, Y. Wu, and J. J. Lim. Demonstration-guided reinforcement learning with learned skills, 2021. URL https://arxiv.org/abs/2107.10253.

[8] K. Hakhamaneshi, R. Zhao, A. Zhan, P. Abbeel, and M. Laskin. Hierarchical few-shot imitation with skill transition models, 2021. URL https://arxiv.org/abs/2107.08981.

[9] K. Pertsch, Y. Lee, and J. J. Lim. Accelerating reinforcement learning with learned skill priors. In *Conference on Robot Learning (CoRL)*, 2020.

[10] K. Pertsch, Y. Lee, and J. J. Lim. Accelerating reinforcement learning with learned skill priors. *arXiv preprint arXiv:2010.11944*, 2020.

[11] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning, 2021. URL https://arxiv.org/abs/1910.11956.

[12] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations, 2017. URL https://arxiv.org/abs/1709.10087.

[13] A. Zhan, P. Zhao, L. Pinto, P. Abbeel, and M. Laskin. A framework for efficient robotic manipulation. *arXiv preprint arXiv:2012.07975*, 2020.

[14] A. Singh, H. Liu, G. Zhou, A. Yu, N. Rhinehart, and S. Levine. Parrot: Data-driven behavioral priors for reinforcement learning, 2020.

[15] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, S. Savarese, and L. Fei-Fei. Roboturk: A crowdsourcing platform for robotic skill learning through imitation, 2018. URL https://arxiv.org/abs/1811.02790.

[16] J. Wong, A. Tung, A. Kurenkov, A. Mandlekar, L. Fei-Fei, S. Savarese, and R. Martín-Martín. Error-aware imitation learning from teleoperation data for mobile manipulation, 2021. URL https://arxiv.org/abs/2112.05251.

[17] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation, 2017. URL https://arxiv.org/abs/1710.04615.

[18] A. Mandlekar, F. Ramos, B. Boots, S. Savarese, L. Fei-Fei, A. Garg, and D. Fox. Iris: Implicit reinforcement without interaction at scale for learning control from offline robot manipulation data. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4414–4420. IEEE, 2020.

[19] S. Cabi, S. Gómez Colmenarejo, A. Novikov, K. Konyushkova, S. Reed, R. Jeong, K. Zolna, Y. Aytar, D. Budden, M. Vecerik, et al. Scaling data-driven robotics with reward sketching and batch reinforcement learning. *arXiv*, pages arXiv–1909, 2019.

[20] A. Mandlekar, D. Xu, R. Martín-Martín, S. Savarese, and L. Fei-Fei. Learning to generalize across long-horizon tasks from human demonstrations. *arXiv preprint arXiv:2003.06085*, 2020.

[21] A. Mandlekar, D. Xu, R. Martín-Martín, Y. Zhu, L. Fei-Fei, and S. Savarese. Human-in-the-loop imitation learning using remote teleoperation, 2020. URL https://arxiv.org/abs/2012.06733.

[22] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer. Hg-dagger: Interactive imitation learning with human experts, 2018. URL https://arxiv.org/abs/1810.02890.

[23] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet. Learning latent plans from play, 2019. URL https://arxiv.org/abs/1903.01973.

[24] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn. Robonet: Large-scale multi-robot learning, 2019. URL https://arxiv.org/abs/1910.11215.

[25] R. Goyal, S. E. Kahou, V. Michalski, J. Materzyńska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic. The "something something" video database for learning and evaluating visual common sense, 2017. URL https://arxiv.org/abs/1706.04261.

[26] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling egocentric vision: The epic-kitchens dataset, 2018. URL https://arxiv.org/abs/1804.02748.

[27] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan, I. Radosavovic, S. K. Ramakrishnan, F. Ryan, J. Sharma, M. Wray, M. Xu, E. Z. Xu, C. Zhao, S. Bansal, D. Batra, V. Cartillier, S. Crane, T. Do, M. Doulaty, A. Erapalli, C. Feichtenhofer, A. Fragomeni, Q. Fu, C. Fuegen, A. Gebreselasie, C. Gonzalez, J. Hillis, X. Huang, Y. Huang, W. Jia, W. Khoo, J. Kolar, S. Kottur, A. Kumar, F. Landini, C. Li, Y. Li, Z. Li, K. Mangalam, R. Modhugu, J. Munro, T. Murrell, T. Nishiyasu, W. Price, P. R. Puentes, M. Ramazanova, L. Sari, K. Somasundaram, A. Southerland, Y. Sugano, R. Tao, M. Vo, Y. Wang, X. Wu, T. Yagi, Y. Zhu, P. Arbelaez, D. Crandall, D. Damen, G. M. Farinella, B. Ghanem, V. K. Ithapu, C. V. Jawahar, H. Joo, K. Kitani, H. Li, R. Newcombe, A. Oliva, H. S. Park, J. M. Rehg, Y. Sato, J. Shi, M. Z. Shou, A. Torralba, L. Torresani, M. Yan, and J. Malik. Ego4d: Around the World in 3,000 Hours of Egocentric Video. In *IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2022.

[28] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation, 2022. URL https://arxiv.org/abs/2203.12601.

[29] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik. Masked visual pre-training for motor control, 2022. URL https://arxiv.org/abs/2203.06173.

[30] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. 2022. URL https://arxiv.org/abs/2202.02005.

[31] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.

[32] C. Lynch and P. Sermanet. Language conditioned imitation learning over unstructured data, 2020. URL https://arxiv.org/abs/2005.07648.

[33] A. Allshire, R. Martín-Martín, C. Lin, S. Manuel, S. Savarese, and A. Garg. Laser: Learning a latent action space for efficient reinforcement learning, 2021. URL https://arxiv.org/abs/2103.15793.

[34] M. Yang, S. Levine, and O. Nachum. Trail: Near-optimal imitation learning with suboptimal data, 2022. URL https://arxiv.org/abs/2110.14770.

[35] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn. Gradient surgery for multi-task learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5824–5836. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/3fe78a8acf5fda99de95303940a2420c-Paper.pdf.

[36] I. Kostrikov, A. Nair, and S. Levine. Offline reinforcement learning with implicit q-learning, 2021. URL https://arxiv.org/abs/2110.06169.

[37] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2020. URL https://arxiv.org/abs/2004.07219.

[38] T. Yu, A. Kumar, Y. Chebotar, K. Hausman, C. Finn, and S. Levine. How to leverage unlabeled data in offline reinforcement learning, 2022. URL https://arxiv.org/abs/2202.01741.