Demonstration-Guided Reinforcement Learning with Learned Skills

Anonymous Author(s)
Affiliation
Address
email

Abstract: Demonstration-guided reinforcement learning (RL) is a promising approach for learning complex behaviors by leveraging both reward feedback and a set of target task demonstrations. Prior approaches for demonstration-guided RL treat every new task as an independent learning problem and attempt to follow the provided demonstrations step-by-step, akin to a human trying to imitate a completely unseen behavior by following the demonstrator's exact muscle movements. Naturally, such learning will be slow, but often new behaviors are not completely unseen: they share subtasks with behaviors we have previously learned. In this work, we aim to exploit this shared subtask structure to increase the efficiency of demonstration-guided RL. We first learn a set of reusable skills from large offline datasets of prior experience collected across many tasks. We then propose **Ski**ll-based **L**earning with **D**emonstrations (**SkiLD**), an algorithm for demonstration-guided RL that efficiently leverages the provided demonstrations by following the demonstrated *skills* instead of the primitive actions, resulting in substantial performance improvements over prior demonstration-guided RL approaches. We validate the effectiveness of our approach on long-horizon maze navigation and complex robot manipulation tasks.

Keywords: Reinforcement Learning, Imitation Learning, Skill-Based Transfer

1 Introduction

Humans are remarkably efficient at acquiring new skills from demonstrations: often a single demonstration of the desired behavior and a few trials of the task are sufficient to master it [1, 2, 3]. To allow for such efficient learning, we can leverage a large number of previously learned behaviors [2, 3]. Instead of imitating precisely each of the demonstrated muscle movements, humans can extract the performed *skills* and use the rich repertoire of already acquired skills to efficiently reproduce the desired behavior.

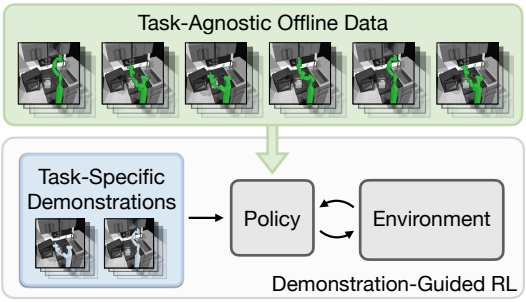

Figure 1: Our approach SkiLD leverages large, task-agnostic datasets collected across many different tasks for efficient demonstration-guided reinforcement learning by (1) acquiring a rich motor skill repertoire from such offline data and (2) understanding and imitating the demonstrations based on the skill repertoire.

Demonstrations are also commonly used in reinforcement learning (RL) to guide exploration and improve sample efficiency [4, 5, 6, 7, 8]. However, such demonstration-guided RL approaches attempt to learn tasks *from scratch*: analogous to a human trying to imitate a completely unseen behavior by following every demonstrated muscle movement, they try to imitate the *primitive actions* performed in the provided demonstrations. As with humans, such step-by-step imitation leads to brittle policies [9], and thus these approaches require many demonstrations and environment interactions to learn a new task.

Submitted to the 5th Conference on Robot Learning (CoRL 2021). Do not distribute.

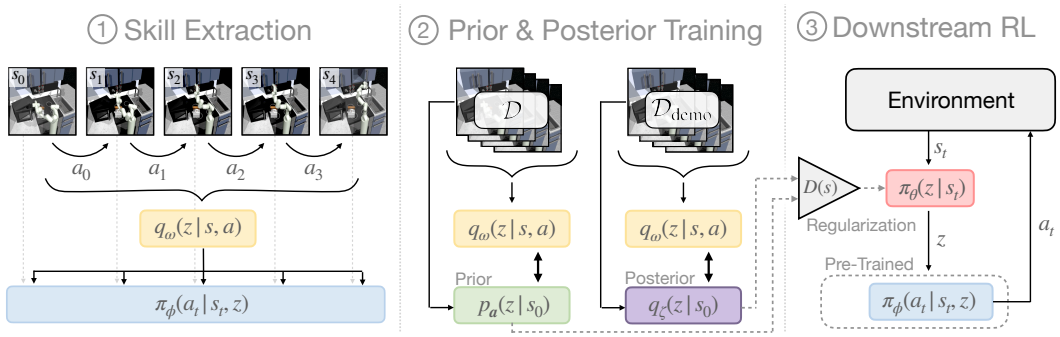

Figure 2: Our approach, SkiLD, combines task-agnostic experience and task-specific demonstrations to efficiently learn target tasks in three steps: (1) extract skill representation from task-agnostic offline data, (2) learn task-agnostic skill prior from task-agnostic data and task-specific skill posterior from demonstrations, and (3) learn a high-level skill policy for the target task using prior knowledge from both task-agnostic offline data and task-specific demonstrations. **Left**: Skill embedding model with skill extractor (**yellow**) and closed-loop skill policy (**blue**). **Middle**: Training of skill prior (**green**) from task-agnostic data and skill posterior (**purple**) from demonstrations. **Right**: Training of high-level skill policy (**red**) on a downstream task using the pre-trained skill representation and regularization via the skill prior and posterior, mediated by the demonstration discriminator $D(s)$.

We propose to improve the efficiency of demonstration-guided RL by leveraging prior experience in the form of an offline "task-agnostic" experience dataset, collected not on one but across many tasks (see Figure 1). Given such a dataset, our approach extracts reusable skills: robust short-horizon behaviors that can be recombined to learn new tasks. Like a human imitating complex behaviors via the chaining of known skills, we can use this repertoire of skills for efficient demonstration-guided RL on a new task by guiding the policy using the demonstrated *skills* instead of the primitive actions.

Concretely, we propose **Ski**ll-based **L**earning with **D**emonstrations (**SkiLD**), a demonstration-guided RL algorithm that learns short-horizon skills from offline datasets and then learns new tasks efficiently by leveraging these skills to follow a given set of demonstrations. Across challenging navigation and robotic manipulation tasks our approach significantly improves the learning efficiency over prior demonstration-guided RL approaches.

In summary, the contributions of our work are threefold: (1) we introduce the problem of leveraging task-agnostic offline datasets for accelerating demonstration-guided RL on unseen tasks, (2) we propose SkiLD, a skill-based algorithm for efficient demonstration-guided RL and (3) we show the effectiveness of our approach on a maze navigation and two complex robotic manipulation tasks.

2 Related Work

Imitation learning and Demonstration-Guided RL. Learning from Demonstration, also known as imitation learning [10], is a common approach for learning complex behaviors by leveraging a set of demonstrations. Demonstrations can also be combined with reinforcement learning [6, 7, 4, 5, 8, 11, 12]. This allows the agent to leverage demonstrations for overcoming exploration challenges in RL while using RL to increase robustness and performance of the imitation learning policies.

Skill-based RL. One class of approaches for leveraging offline datasets that is particularly suited for learning long-horizon behaviors is skill-based RL [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23]. These methods extract reusable skills from task-agnostic datasets and learn new tasks by recombining them. In our work we combine the best of both worlds: by using large, task-agnostic datasets and a small number of task-specific demonstrations, we accelerate the learning of long-horizon tasks while reducing the number of required demonstrations.

For a more detailed discussion of related work, see appendix, section H.

3 Approach

Our goal is to use skills extracted from task-agnostic prior experience data to improve the efficiency of demonstration-guided RL on a new task. We aim to leverage a set of provided demonstrations by following the performed *skills* as opposed to the primitive actions. Therefore, we need a model that can (1) leverage prior data to learn a rich set of skills and (2) identify the skills performed in the demonstrations in order to follow them. Next, we formally define our problem, summarize relevant prior work on RL with learned skills and then describe our demonstration-guided RL approach.

3.1 Preliminaries

Problem Formulation We assume access to two types of datasets: a large task-agnostic offline dataset and a small task-specific demonstration dataset. The task-agnostic dataset $\mathcal{D} = \{s_t, a_t, ...\}$ consists of trajectories of meaningful agent behaviors, but includes no demonstrations of the target task. We only assume that its trajectories contain *short-horizon* behaviors that can be reused to solve the target task. Such data can be collected without a particular task in mind using a mix of sources, e.g., via human teleoperation, autonomous exploration, or through policies trained for other tasks. Since it can be used to accelerate *many* downstream task that utilize similar short-term behaviors we call it *task-agnostic*. In contrast, the task-specific data is a much smaller set of demonstration trajectories $\mathcal{D}_{\text{demo}} = \{s_t^d, a_t^d, ...\}$ that are specific to a single target task.

The downstream learning problem is formulated as a Markov decision process (MDP) defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \rho, \gamma)$ of states, actions, transition probabilities, rewards, initial state distribution, and discount factor. We aim to learn a policy $\pi_\theta(a|s)$ with parameters θ that maximizes the discounted sum of rewards $J(\theta) = \mathbb{E}_\pi\big[\sum_{t=0}^{T-1} J_t\big] = \mathbb{E}_\pi\big[\sum_{t=0}^{T-1} \gamma^t r_t\big]$, where T is the episode horizon.

Skill Prior RL Our goal is to extract skills from task-agnostic experience data and reuse them for *demonstration-guided* RL. Prior work has investigated the reuse of learned skills for accelerating RL [23]. In this section, we will briefly summarize their proposed approach Skill Prior RL (SPiRL) and then describe how our approach improves upon it in the *demonstration-guided* RL setting.

SPiRL defines a skill as a sequence of H consecutive actions $\boldsymbol{a} = \{a_t, \dots, a_{t+H-1}\}$, where the skill horizon H is a hyperparameter. It uses the task-agnostic data to jointly learn (1) a generative model of skills $p(\boldsymbol{a}|z)$, that decodes latent skill embeddings z into executable action sequences $\boldsymbol{a}$, and (2) a state-conditioned prior distribution $p(z|s)$ over skill embeddings. For learning a new downstream task, SPiRL trains a high-level skill policy $\pi_\theta(z|s)$ whose outputs get decoded into executable actions using the pre-trained skill decoder. Crucially, the learned skill prior is used to guide the policy during downstream RL by maximizing the following divergence-regularized RL objective:

$$J(\theta) = \mathbb{E}_{\pi_\theta}\Bigg[\sum_{t=0}^{T-1} r(s_t, z_t) - \alpha D_{\text{KL}}\big(\pi_\theta(z_t|s_t), p(z_t|s_t)\big)\Bigg]. \tag{1}$$

Here, the KL-divergence term ensures that the policy remains close to the learned skill prior, guiding exploration during RL. By combining this guided exploration with temporal abstraction via the learned skills, SPiRL substantially improves the efficiency of RL on long-horizon tasks.

3.2 Skill Representation Learning

We leverage SPiRL's skill embedding model for learning our skill representation. We follow prior work on skill-based RL [21, 24] and increase the expressiveness of the skill representation by replacing SPiRL's low-level skill decoder $p(\boldsymbol{a}|z)$ with a closed-loop skill policy $\pi_\phi(a|s, z)$ with parameters ϕ that is conditioned on the current environment state. In our experiments we found this closed-loop decoder to improve performance (see Section C for an empirical comparison).

Figure 2 (left) summarizes our skill learning model. It consists of two parts: the skill inference network $q_\omega(z|s_{0:H-1}, a_{0:H-2})$ with parameters ω and the closed-loop skill policy $\pi_\phi(a_t|s_t, z_t)$. Note that in contrast to SPiRL the skill inference network is state-conditioned to account for the state-conditioned low-level policy. During training we randomly sample an H-step state-action trajectory from the task-agnostic dataset and pass it to the skill inference network, which predicts the low-dimensional skill embedding z. This skill embedding is then input into the low-level policy

$\pi_\phi(a_t|s_t, z)$ for every input state. The policy is trained to imitate the given action sequence, thereby learning to reproduce the behaviors encoded by the skill embedding z.

The latent skill representation is optimized using variational inference, which leads to the full skill learning objective:

$$\max_{\phi,\omega} \mathbb{E}_q \Bigg[\underbrace{\prod_{t=0}^{H-2} \log \pi_\phi(a_t|s_t, z)}_{\text{behavioral cloning}} - \beta \big(\underbrace{\log q_\omega(z|s_{0:H-1}, a_{0:H-2}) - \log p(z)}_{\text{embedding regularization}} \big) \Bigg]. \tag{2}$$

We use a unit Gaussian prior $p(z)$ and weight the embedding regularization term with a factor β [25].

3.3 Demonstration-Guided RL with Learned Skills

To leverage the learned skills for accelerating demonstration-guided RL on a new task, we use a hierarchical policy learning scheme (see Figure 2, right): a high-level policy $\pi_\theta(z|s)$ outputs latent skill embeddings z that get decoded into actions using the pre-trained low-level skill policy. We freeze the weights of the skill policy during downstream training for simplicity.[1]

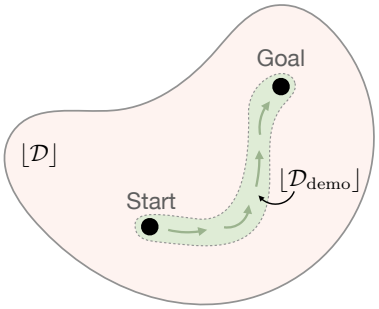

Figure 3: We leverage prior experience data $\mathcal{D}$ and demonstration data D_{demo}. Our policy is guided by the task-specific skill posterior $q_\zeta(z|s)$ within the support of the demonstrations (**green**) and by the task-agnostic skill prior $p_{\boldsymbol{a}}(z|s)$ otherwise (**red**). The agent also receives a reward bonus for reaching states in the demonstration support.

Our goal is to leverage the task-specific demonstrations to guide learning of the high-level policy on the new task. In Section 3.1, we showed how SPiRL [23] leverages a learned *skill prior* $p(z|s)$ to guide exploration. However, this prior is task-agnostic, i.e., it encourages exploration of *all* skills that are meaningful to be explored, independent of which task the agent is trying to solve. Even though SPiRL's objective makes learning with a large number of skills more efficient, it encourages the policy to explore many skills that are not relevant to the downstream task.

In this work, we propose to extend the skill prior guided approach and leverage target task demonstrations to additionally learn a *task-specific* skill distribution, which we call *skill posterior* $q_\zeta(z|s)$ with parameters ζ (in contrast to the skill prior it is conditioned on the target task, hence "posterior"). We train this skill posterior by using the pre-trained skill inference model $q_\omega(z|s_{0:H-1}, a_{0:H-2})$ to extract the embeddings for the skills performed in the demonstration sequences (see Figure 2, middle):

$$\min_\zeta \mathbb{E}_{(s,a)\sim\mathcal{D}_{\text{demo}}} D_{\text{KL}}\big(q_\omega(z|s_{0:H-1}, a_{0:H-2}), q_\zeta(z|s_0)\big), \tag{3}$$

where D_{KL} denotes the Kullback-Leibler divergence.

A naive approach for leveraging the skill posterior is to simply use it to replace the skill prior in Equation 1, i.e., to regularize the policy to stay close to the skill posterior in every state. However, the trained skill posterior is only accurate within the support of the demonstration dataset $\lfloor \mathcal{D}_{\text{demo}} \rfloor$, because by definition it was only trained on demonstration sequences. Since $|\mathcal{D}_{\text{demo}}| \ll |\mathcal{D}|$, this support will only be a small subset of all states (see Figure 3) and thus the skill posterior will often provide incorrect guidance in states outside the demonstrations' support.

Instead, we propose to use a three-part objective to guide the policy during downstream learning. Our goal in formulating this objective is to (1) follow the skill posterior *within* the support of the demonstrations, (2) follow the skill prior *outside* the demonstration support, and (3) encourage the policy to reach states *within* the demonstration support. Crucial for all three components is to determine whether a given state is within the support of the demonstration data. We propose to use a learned discriminator $D(s)$ to answer this question. $D(s)$ is a binary classifier that distinguishes demonstration and non-demonstration states and it is trained using samples from the demonstration and task-agnostic datasets, respectively.

[1] Joint optimization of high-level and low-level policy is possible, but we did not find it necessary in our experiments. Prior work on hierarchical RL [26] suggests that joint optimization can lead to training instabilities. We will leave an investigation of this for future work.

In summary, our algorithm pre-trains the following components:

$$\begin{aligned} \text{low-level skill policy:} \quad & \pi_\phi(a|s,z) \\ \text{task-agnostic skill prior:} \quad & p(z|s) \\ \text{task-specific skill posterior:} \quad & q_\zeta(z|s) \\ \text{demo vs non-demo discriminator:} \quad & D(s) \end{aligned}$$

Note that the first two components can be reused for arbitrary target tasks. Thus, only the latter two components need to be re-trained for a new target task using the demonstration data. Since all pre-training is done offline without the need for e.g. adversarial training or training in the RL loop, we find that pre-training converges stably across a wide range of architecture and hyperparameter choices and results can be separately visualized and debugged.

Once all components are pre-trained, we use the discriminator's output to weight terms in our objective that regularize the high-level policy $\pi_\theta(z|s)$ towards the skill prior or posterior. Additionally, we provide a reward bonus for reaching states which the discriminator classifies as being within the demonstration support. This results in the following term J_t for SkiLD's full RL objective:

$$\begin{aligned} J_t = \tilde{r}(s_t, z_t) \underbrace{-\alpha_q D_{\text{KL}}(\pi_\theta(z_t|s_t), q_\zeta(z_t|s_t)) \cdot D(s_t)}_{\text{posterior regularization}} \underbrace{-\alpha D_{\text{KL}}(\pi_\theta(z_t|s_t), p(z_t|s_t)) \cdot (1 - D(s_t))}_{\text{prior regularization}}, \\ \text{with } \tilde{r}(s_t, z_t) = (1-\kappa) \cdot r(s_t, z_t) \underbrace{+ \kappa \cdot \big[\log D(s_t) - \log\big(1 - D(s_t)\big)\big]}_{\text{discriminator reward}}. \end{aligned} \tag{4}$$

The weighting factor κ is a hyperparameter; α and α_q are either constant or tuned automatically via dual gradient descent [27]. The discriminator reward follows common formulations used in adversarial imitation learning [28, 29, 8, 30].[2] Our formulation combines IRL-like and BC-like objectives by using learned rewards *and* trying to match the demonstration's skill distribution.

For policy optimization, we use a modified version of the SPiRL algorithm [23], which itself is based on Soft Actor-Critic [31]. Concretely, we replace the environment reward with the discriminator-augmented reward and all prior divergence terms with our new, weighted prior-posterior-divergence terms from equation 4 (for the full algorithm see appendix, Section A).

4 Experiments

In this paper, we propose to leverage a large offline experience dataset for efficient demonstration-guided RL. We aim to answer the following questions: (1) Can the use of task-agnostic prior experience improve the efficiency of *demonstration-guided* RL? (2) Does the reuse of pre-trained skills reduce the number of required target-specific demonstrations? (3) In what scenarios does the combination of prior experience and demonstrations lead to the largest efficiency gains?

4.1 Experimental Setup and Comparisons

To evaluate whether our method SkiLD can efficiently use task-agnostic data, we compare it to prior demonstration-guided RL approaches on three complex, long-horizon tasks: a 2D maze navigation task, a robotic kitchen manipulation task and a robotic office cleaning task (see Figure 4, left).

Maze Navigation. We adapt the maze navigation task from Pertsch et al. [23] and increase task complexity by adding randomness to the agent's initial position. The agent needs to navigate through the maze to a fixed goal position using planar velocity commands. It only receives a binary reward upon reaching the goal and its observations consist of its 2D position and velocity. This environment is challenging for prior demonstration-guided RL approaches since demonstrations of the task span hundreds of time steps, making step-by-step imitation of primitive actions inefficient. We collect a task-agnostic offline experience dataset with 3000 sequences using a motion planner to find paths between randomly sampled start-goal pairs. This data can be used to extract relevant short-horizon skills like navigating hallways or passing through narrow doors. For the target task we sample an unseen start-goal pair and collect 5 demonstrations for reaching the goal position from states sampled around the start position.

[2] We found that using the pre-trained discriminator weights led to stable training, but it is possible to perform full adversarial training by finetuning $D(s)$ with rollouts from the downstream task training. We report results for initial experiments with discriminator finetuning in Section E and leave further investigation for future work.

Robot Kitchen Environment. We use the environment of Gupta et al. [19] in which a 7DOF robot arm needs to perform a sequence of four subtasks, such as opening the microwave or switching on the light, in the correct order. The agent receives a binary reward upon completion of each consecutive subtask. Following Gupta et al. [19] the agent's 30-dimensional observation consists of its joint velocities as well as low-dimensional representations of state of movable objects, like their positions or opening angles. It outputs 9-dimensional action commands, including two continuous closing actions for the gripper fingers. In addition to the long task horizon, this environment requires precise control of a high-DOF manipulator, testing the scalability of our approach. We use 603 teleoperated sequences performing various subtask combinations (from Gupta et al. [19]) as our task-agnostic experience dataset $\mathcal{D}$ and separate a set of 20 demonstrations for one particular sequence of subtasks, which we define as our target task (see Figure 4, middle). From the offline data the agent learns basic kitchen manipulation skills like opening the microwave or turning on the stove, which need to be recombined in novel ways to solve the target task.

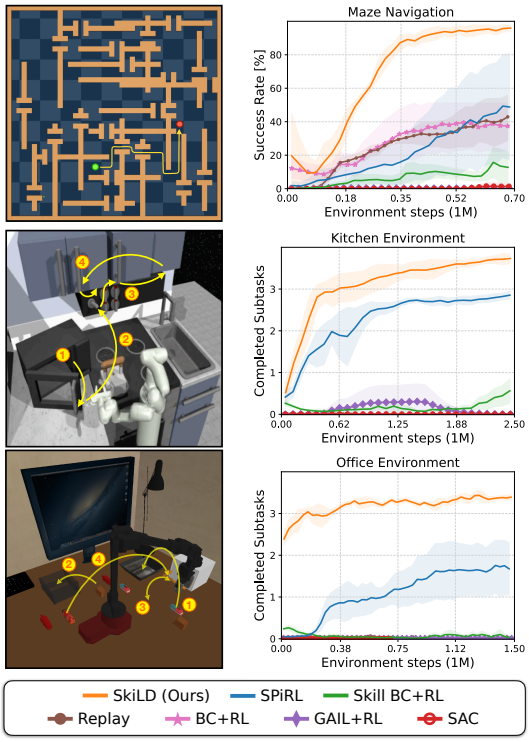

Figure 4: **Left**: Test environments, **top to bottom**: 2D maze navigation, robotic kitchen manipulation and robotic office cleaning. **Right**: Target task performance vs environment steps. By using task-agnostic experience, our approach more efficiently leverages the demonstrations than prior demonstration-guided RL approaches across all tasks. The comparison to SPiRL shows that demonstrations improve efficiency even if the agent has access to large amounts of prior experience.

Robot Office Environment. A 5 DOF robot arm needs to clean an office environment by placing objects in their target bins or putting them in a drawer. It receives binary rewards for the completion of each subtask. The 97-dimensional observation consists of the position and orientation of the agent's end-effector and all movable objects in the scene. The agent controls an 8-dimensional action space of end-effector position and orientation and a continuous gripper closing action. In addition to the challenges of the kitchen environment, this task tests the ability of our approach to learn long-horizon behaviors with freely manipulatable objects. We collect 2400 training trajectories by placing the objects at randomized positions in the environment and performing random subtasks using scripted policies. From this, the agent learns general object manipulation skills like picking, placing and opening or closing drawers. We also collect 50 demonstrations for the unseen target task with new object locations and subtask sequence.

We compare our approach to multiple prior demonstration-guided RL approaches that represent the different classes of existing algorithms introduced in Section 2. In contrast to SkiLD, these approaches are not designed to leverage task-agnostic prior experience: **BC + RL** initializes a policy with behavioral cloning of the demonstrations, then continues to apply BC loss while finetuning the policy with Soft Actor-Critic (SAC, [31]), representative of [6, 7]. **GAIL + RL** [8] combines rewards from the environment and adversarial imitation learning (GAIL, [32]), and optimizes the policy using PPO [33]. **Demo Replay** initializes the replay buffer of an SAC agent with the demonstrations and uses them with prioritized replay during updates, representative of [4]. We also compare our approach to RL-only methods to show the benefit of using demonstration data: **SAC** [31] is a state-of-the-art model-free RL algorithm, it neither uses offline experience nor demonstrations. **SPiRL** [23] extracts skills from task-agnostic experience and performs prior-guided RL on the target task (see Section 3.1)[3]. Finally, we compare to a baseline that combines skills learned from task-agnostic data with target task demonstrations: **Skill BC+RL** encodes the demonstrations with the pre-trained

[3]We train SPiRL with the closed-loop policy representation from Section 3.2 for fair comparison and better performance. For an empirical comparison of open and closed-loop skill representations in SPiRL, see Section C.

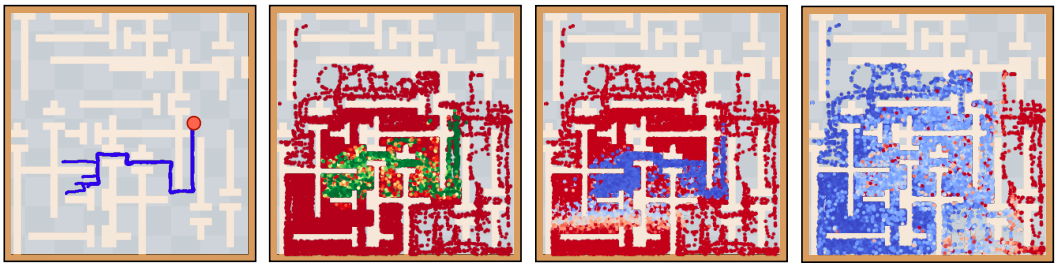

Figure 5: Visualization of our approach on the maze navigation task (visualization states collected by rolling out the skill prior). **Left**: the given demonstration trajectories; **Middle left**: output of the demonstration discriminator $D(s)$ (the **greener**, the higher the predicted probability of a state to be within demonstration support, **red** indicates low probability). **Middle right**: policy divergences to the skill posterior and **Right**: divergence to the skill prior (**blue** indicates small and **red** high divergence). The discriminator accurately infers the demonstration support, the policy successfully follows the skill posterior only within the demonstration support and the skill prior otherwise.

skill encoder and runs BC for the high-level skill policy, then finetunes on the target task using SAC. For further details on the environments, data collection, and implementation, see appendix Section B.

4.2 Demonstration-Guided RL with Learned Skills

Maze Navigation. We compare the downstream task performance of the tested methods on the maze navigation task in Figure 4 (right). Prior approaches for demonstration-guided RL struggle to learn the task since task-rewards are sparse and only five demonstrations are provided. With such small coverage, behavioral cloning of the demonstrations' primitive actions leads to brittle policies which are hard to finetune (for an analysis of the influence of the number of demonstrations, see Section 4.3). The Replay agent improves over SAC without demonstrations and partly succeeds at the task, but learning is slow. The GAIL+RL approach is able to follow part of the demonstrated behavior, but fails to reach the final goal and as a result does not receive the sparse environment reward (see Figure 8 for qualitative results). SPiRL and Skill BC+RL, in contrast, are able to leverage offline, task-agnostic experience to learn to occasionally solve the task, but require a substantial amount of environment interactions: SPiRL's learned, task-agnostic skill prior and Skill BC+RL's uniform skill prior during SAC finetuning encourage the policy to try many skills before converging to the ones that solve the downstream task[4]. In contrast, our approach SkiLD leverages the task-specific skill posterior to quickly explore the relevant skills, leading to significant efficiency gains (see Figure 9 for a comparison of the exploration of SkiLD vs. SPiRL). We qualitatively analyze our approach in Figure 5.

Robotic Manipulation. We show the performance comparison on the robotic manipulation tasks in Figure 4 (right)[5]. Both tasks are more challenging since they require precise control of a high-DOF manipulator. We find that approaches for demonstration-guided RL that do not leverage task-agnostic experience struggle to learn either of the tasks since following the demonstrations step-by-step is inefficient and prone to accumulating errors. SPiRL, in contrast, is able to learn meaningful skills from the offline datasets, but struggles to explore the task-relevant skills and therefore learns slowly. Worse yet, the uniform skill prior used in Skill BC+RL's SAC finetuning is even less suited for the target task and leads the policy to deviate from the BC initialization early on in training, preventing the agent from learning the task altogether (for pure BC performance, see appendix, Figure 13). Our approach, however, uses the learned skill posterior to guide the chaining of the extracted skills and thereby learns to solve the tasks efficiently, showing how SkiLD effectively combines task-agnostic and task-specific data for demonstration-guided RL.

[4]Performance of SPiRL differs from Pertsch et al. [23] due to increased task complexity, see Section B.4.
[5]For qualitative robot manipulation videos, see `https://sites.google.com/view/skill-demo-rl`.

4.3 Ablation Studies

In Figure 6 (left) we test the robustness of our approach to the **number of demonstrations** in the maze navigation task and compare to BC+RL, which we found to work best across different demonstration set sizes. Both approaches benefit from more demonstrations, but our approach is able to learn with much fewer demonstrations by using prior experience. While BC+RL learns each low-level action from the demonstrations, SkiLD merely learns to recombine skills it has already mastered using the offline data, thus requiring less dense supervision and fewer demonstrations. We also ablate the **components of our RL objective** on the kitchen task in Figure 6 (right). Removing the discriminator reward bonus ("*no-GAIL*") slows convergence since the agent lacks a dense reward signal. Naively replacing the skill prior in the SPiRL objective of Equation 1 with the learned skill *posterior* ("*post-only*") fails since the agent follows the skill posterior outside its support. Removing the skill posterior and optimizing a discriminator bonus augmented reward using SPiRL ("*no-post*") fails because the agent cannot efficiently explore the rich skill space. Finally, we show the efficacy of our approach in the pure imitation setting, without environment rewards, in appendix, Section E.

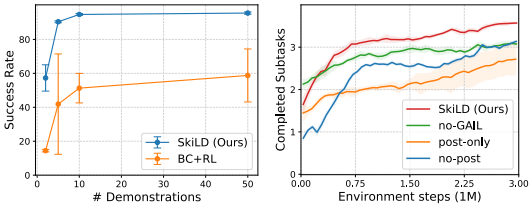

Figure 6: Ablation studies. We test the performance of SkiLD for different sizes of the demonstration dataset $|\mathcal{D}_{\text{demo}}|$ on the maze navigation task (**left**) and ablate the components of our objective on the kitchen manipulation task (**right**).

4.4 Data Alignment Analysis

We aim to analyze in what scenarios the use of demonstrations *in addition* to task-agnostic experience is most beneficial. In particular, we evaluate how the alignment between the distribution of observed behaviors in the task-agnostic dataset and the target behaviors influences learning efficiency. We choose two different target tasks in the kitchen environment, one with good and one with bad alignment between the behavior distributions, and compare our method, which uses demonstrations, to SPiRL, which only relies on the task-agnostic data[6].

In the **well-aligned** case (Figure 7, solid lines), we find that both approaches learn the task efficiently. Since the skill prior encourages effective exploration on the downstream task, the benefit of the additional demonstrations leveraged in our method is marginal. In contrast, if task-agnostic data and downstream task are **not well-aligned** (Figure 7, dashed), SPiRL struggles to learn the task since it cannot maximize task reward and minimize divergence from the mis-aligned skill prior at the same time. Our approach learns more reliably by encouraging the policy to reach demonstration-like states and then follow the skill posterior, which by design is well-aligned with the target task.

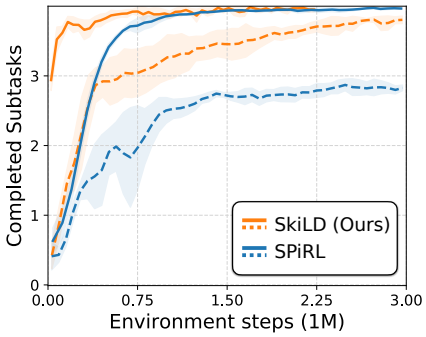

Figure 7: Analysis of data vs. task alignment. The benefit of using demonstrations *in addition* to prior experience diminishes if the prior experience is closely aligned with the target task (**solid**), but gains are high when data and task are not well-aligned (**dashed**).

In summary, our analysis finds that approaches which leverage both task-agnostic data *and* demonstrations, improve over methods that use either of the data sources alone across all tested tasks. We find that combining the data sources is particularly beneficial in two cases:

- **Diverse Task-Agnostic Data.** Demonstrations can focus exploration on task-relevant skills if the task-agnostic skill prior explores a too large set of skills (see Section 4.2).
- **Mis-Aligned Task-Agnostic Data.** Demonstrations can compensate mis-alignment between task-agnostic data and target task by guiding exploration with the skill posterior instead of the mis-aligned prior.

[6]For a detailed analysis of the behavior distributions in the kitchen dataset and the chosen tasks, see Section F.

References

[1] H. Bekkering, A. Wohlschlager, and M. Gattis. Imitation of gestures in children is goal-directed. *The Quarterly Journal of Experimental Psychology: Section A*, 53(1):153–164, 2000.

[2] S. A. Al-Abood, K. Davids, and S. J. Bennett. Specificity of task constraints and effects of visual demonstrations and verbal instructions in directing learners' search during skill acquisition. *Journal of motor behavior*, 33(3):295–305, 2001.

[3] N. J. Hodges, A. M. Williams, S. J. Hayes, and G. Breslin. What is modelled during observational learning? *Journal of sports sciences*, 25(5):531–545, 2007.

[4] M. Vecerik, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, and M. Riedmiller. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*, 2017.

[5] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, et al. Deep q-learning from demonstrations. In *AAAI*, 2018.

[6] 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. In *Robotics: Science and Systems*, 2018.

[7] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6292–6299. IEEE, 2018.

[8] Y. Zhu, Z. Wang, J. Merel, A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramár, R. Hadsell, N. de Freitas, and N. Heess. Reinforcement and imitation learning for diverse visuomotor skills. In *Robotics: Science and Systems*, 2018.

[9] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*, pages 627–635, 2011.

[10] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[11] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018.

[12] J. Merel, Y. Tassa, D. TB, S. Srinivasan, J. Lemmon, Z. Wang, G. Wayne, and N. Heess. Learning human behaviors from motion capture by adversarial imitation. *arXiv preprint arXiv:1707.02201*, 2017.

[13] K. Hausman, J. T. Springenberg, Z. Wang, N. Heess, and M. Riedmiller. Learning an embedding space for transferable robot skills. In *ICLR*, 2018.

[14] J. Merel, L. Hasenclever, A. Galashov, A. Ahuja, V. Pham, G. Wayne, Y. W. Teh, and N. Heess. Neural probabilistic motor primitives for humanoid control. *ICLR*, 2019.

[15] T. Kipf, Y. Li, H. Dai, V. Zambaldi, E. Grefenstette, P. Kohli, and P. Battaglia. Compositional imitation learning: Explaining and executing one task at a time. *ICML*, 2019.

[16] J. Merel, S. Tunyasuvunakool, A. Ahuja, Y. Tassa, L. Hasenclever, V. Pham, T. Erez, G. Wayne, and N. Heess. Catch & carry: Reusable neural controllers for vision-guided whole-body tasks. *ACM. Trans. Graph.*, 2020.

[17] T. Shankar, S. Tulsiani, L. Pinto, and A. Gupta. Discovering motor programs by recomposing demonstrations. In *ICLR*, 2019.

[18] W. Whitney, R. Agarwal, K. Cho, and A. Gupta. Dynamics-aware embeddings. *ICLR*, 2020.

[19] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *CoRL*, 2019.

[20] Y. Lee, J. Yang, and J. J. Lim. Learning to coordinate manipulation skills via skill behavior diversification. In *ICLR*, 2020.

[21] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet. Learning latent plans from play. In *CoRL*, 2020.

[22] K. Pertsch, O. Rybkin, J. Yang, S. Zhou, K. Derpanis, J. Lim, K. Daniilidis, and A. Jaegle. Keyframing the future: Keyframe discovery for visual prediction and planning. *L4DC*, 2020.

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

[24] 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.

[25] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[26] A. Levy, G. Konidaris, R. Platt, and K. Saenko. Learning multi-level hierarchies with hindsight. *ICLR*, 2019.

[27] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[28] C. Finn, P. Christiano, P. Abbeel, and S. Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *NeurIPS Workshop on Adversarial Training*, 2016.

[29] J. Fu, K. Luo, and S. Levine. Learning robust rewards with adversarial inverse reinforcement learning. In *ICLR*, 2018.

[30] I. Kostrikov, K. K. Agrawal, D. Dwibedi, S. Levine, and J. Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. In *ICLR*, 2019.

[31] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *ICML*, 2018.

[32] J. Ho and S. Ermon. Generative adversarial imitation learning. *NeurIPS*, 2016.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[34] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han. On the variance of the adaptive learning rate and beyond. In *ICLR*, 2020.

[35] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[36] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

[37] A. Singh, A. Yu, J. Yang, J. Zhang, A. Kumar, and S. Levine. Cog: Connecting new skills to past experience with offline reinforcement learning. *CoRL*, 2020.

[38] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, pages 305–313, 1989.

[39] 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. *CoRL*, 2019.

[40] S. Cabi, S. G. Colmenarejo, A. Novikov, K. Konyushkova, S. Reed, R. Jeong, K. Zolna, Y. Aytar, D. Budden, M. Vecerik, O. Sushkov, D. Barker, J. Scholz, M. Denil, N. de Freitas, and Z. Wang. Scaling data-driven robotics with reward sketching and batch reinforcement learning. *RSS*, 2019.

[41] 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. In *CoRL*, 2018.

[42] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *ICML*, 2004.

[43] N. Y. Siegel, J. T. Springenberg, F. Berkenkamp, A. Abdolmaleki, M. Neunert, T. Lampe, R. Hafner, and M. Riedmiller. Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. *ICLR*, 2020.

[44] A. Nair, M. Dalal, A. Gupta, and S. Levine. Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

[45] A. Singh, H. Liu, G. Zhou, A. Yu, N. Rhinehart, and S. Levine. Parrot: Data-driven behavioral priors for reinforcement learning. *ICLR*, 2021.

[46] A. Sharma, S. Gu, S. Levine, V. Kumar, and K. Hausman. Dynamics-aware unsupervised discovery of skills. *ICLR*, 2020.