Skill-based Model-based Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract: Model-based reinforcement learning (RL) is a sample-efficient way of learning complex behaviors by leveraging a learned dynamics model to optimize a policy and plan actions in imagination. Prior work in model-based RL has been mostly confined to using a single-step dynamics model, akin to a human paying attention to every single muscle movement for planning. Instead, humans plan with high-level skills (i.e. temporal abstraction of primitive actions) to solve long-horizon tasks. In this work, we present a **Ski**ll-based **Mo**del-based RL framework (**SkiMo**) that enables planning in the skill space using a *skill dynamics model*, which directly predicts the skill outcomes, rather than step-by-step predicting all small details in the intermediate states. For accurate and efficient long-term planning, we *jointly* learn the skill dynamics model and a skill repertoire from large prior experience. We then harness the learned skill dynamics model to accurately simulate and plan over long horizons in the skill space, which enables efficient downstream learning of long-horizon, sparse reward tasks. Experimental results in navigation and manipulation domains show that SkiMo extends the temporal horizon of model-based approaches and improves the sample efficiency for both model-based RL and skill-based RL.

Keywords: Model-Based Reinforcement Learning, Skill Dynamics Model

1 Introduction

A key trait of human intelligence is the ability to plan abstractly for solving complex tasks [1]. For instance, we perform cooking by imagining outcomes of high-level skills like washing and cutting vegetables, instead of planning every muscle movement involved [2]. This ability to plan with temporally extended skills helps to scale the efficiency of our internal model to long-horizon tasks by reducing the search space of behaviors. To apply this insight to artificial agents, we propose a novel skill-based and model-based hierarchical reinforcement learning (RL) method. We jointly learn a model and a policy in a skill space, which are then used together to learn a complex task, enabling accurate long-term prediction and efficient long-term planning.

Our work is the first to combine model-based RL and skill-based RL to achieve sample efficiency and long-horizon decision making. Typically, model-based RL involves learning a flat (single-step) dynamics model, which predicts the next state from the current state and action. This model can then be used to simulate "imaginary" trajectories for planning to demonstrate significant sample efficiency gains over their model-free alternatives [3, 4]. However, such model-based RL approaches have shown only limited success in long-horizon tasks due to low long-horizon prediction accuracy [5] and computationally expensive search [6, 7, 8].

Skill-based RL enables agents to solve long-horizon tasks by acting with multi-action subroutines, or skills [9, 10, 11, 12, 13, 14], instead of primitive actions at every time step. This temporal abstraction of actions enables systematic long-range exploration and allows RL agents to plan farther into the future, while requiring a shorter horizon for policy optimization. This makes long-horizon downstream tasks more tractable. Yet, on complex long-horizon tasks, such as furniture assembly [15], skill-based RL still requires a few million to billion environment interactions to learn [13], which is impractical for real-world applications (e.g. robotics and healthcare).

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

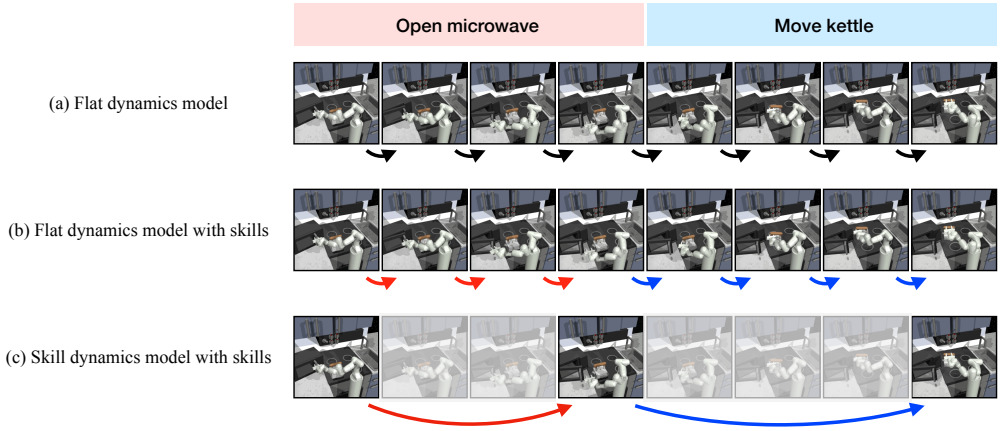


Figure 1: Intelligent agents can use their internal models to imagine potential futures for planning. Instead of planning out every primitive action (black arrows in **a**), they aggregate action sequences into skills (red and blue arrows in **b**). Further, instead of simulating each low-level step, they can leap directly to the predicted outcomes of executing skills in sequence (red and blue arrows in **c**).

To combine the best of both model-based RL and skill-based RL, we propose **Ski**ll-based **Mo**del-based RL (**SkiMo**). Its key component is a *skill dynamics model* for effective planning in the skill space. Given a state and a skill to execute, the skill dynamics model directly predicts the resultant state after skill execution (instead of after one action like the single-step model). Learning a skill dynamics model allows the agent to forecast the eventual consequence of the skill without needing to model every intermediate step and low-level action (Figure 1). Thus, planning with skill dynamics requires fewer predictions to evaluate a plan's outcome than flat, single-step dynamics, resulting in more reliable long-term future predictions and plans.

Concretely, we first jointly learn the skill dynamics model and a skill repertoire from large offline datasets collected across diverse tasks [16, 12, 17]. This joint training shapes the skill embedding space for easy skill dynamics prediction and skill execution. Then, to solve a complex downstream task, we train a hierarchical task policy that acts on the learned skill space. We leverage the learned skill dynamics model to simulate skill trajectories for more sample efficiency policy learning.

The main contribution of this work is to propose *Skill-based Model-based RL (SkiMo)*, a sample-efficient model-based hierarchical RL algorithm that leverages task-agnostic data to extract not only a reusable skill set but also the skill dynamics model. The skill dynamics model enables us to do efficient and accurate long-term planning for sample efficient RL. Our experiments show that our method outperforms the state-of-the-art skill-based and model-based RL algorithms on long-horizon navigation and robotic manipulation tasks with sparse rewards.

2 Related Work

Model-based RL leverages a learned model of the environment to plan a sequence of actions ahead of time that leads to the desired behavior. The model predicts the state of the environment, and optionally the associated reward, after taking a specific action for planning [3, 8, 4] or subsequent policy search [18, 3, 19, 4]. Through simulating candidate lines of behaviors in imagination instead of in the physical environment, model-based algorithms improve the sample efficiency of RL agents on various benchmarks [3, 4]. Conventionally model-based approaches use either Cross-Entropy Method (CEM [20]) or Model-Predictive Path Integral (MPPI [21]) for planning with population-based search over the current model [22, 23, 24]. Instead of planning, Dreamer [3] learns a policy based on the predicted value of imagined states. However, due to the accumulation of prediction error at each step [5] and the large search space for long-horizon planning, the search for an optimal plan on long-horizon tasks using model-based RL is inaccurate and computationally expensive [6, 7, 8].

To facilitate learning of long-horizon behaviors, many works in skill-based RL have proposed to let the agent act over temporally extended skills (i.e. options [25] or motion primitives [26]), which can be represented as sub-policies or a coordinated sequence of low-level actions [25]. Temporal

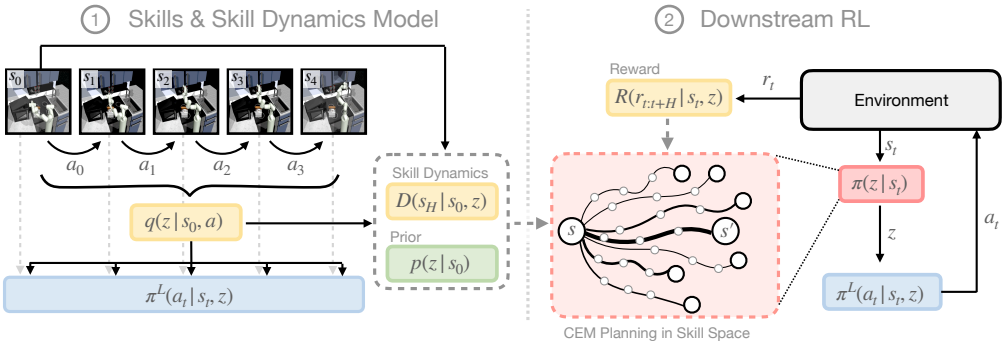


Figure 2: Our approach, SkiMo, combines model-based RL and skill-based RL for sample efficient learning of long-horizon tasks. SkiMo consists of two phases: (1) learn a skill dynamics model and a skill repertoire from offline task-agnostic data, and (2) learn a high-level policy for the downstream task by leveraging the learned model and skills.

abstraction effectively reduces the task horizon for the agent and enables directed exploration [27], a major challenge in RL. However, on complex tasks such as furniture assembly [15], skill-based RL still requires a few million to billions of environment interactions to learn [13], which is impractical for real-world applications. In this paper, we use model-based RL to guide the planning of skills to improve the sample efficiency of skill-based approaches.

There have been attempts to plan over skills in model-based RL [28, 5, 29, 30]. However, these approaches still utilize the conventional single-step dynamics model, which struggles at handling long-horizon planning due to error accumulation. To fully unleash the potential of temporally abstracted skills, we devise a skill-level dynamics model to provide accurate long-term prediction, which is essential for solving long-horizon tasks. To the best of our knowledge, SkiMo is the first work that jointly learning skills and a skill dynamics model from data for model-based RL.

3 Method

To enable accurate long-term prediction and efficient long-horizon planning for sample efficient long-horizon RL, we introduce *SkiMo*, a novel skill-based and model-based RL method that combines the benefits of both frameworks. A key change to prior model-based approaches is the use of a *skill dynamics model* that directly predicts the outcome of a chosen skill, which enables efficient and accurate long-term prediction. We describe the overview of our approach in Section 3.2, which consists of two phases: (1) learning the skill dynamics model and skills from an offline task-agnostic dataset (Section 3.2.2) and (2) downstream task learning with the skill dynamics model (Section 3.2.3).

3.1 Preliminaries

Reinforcement Learning: We formulate our problem as a Markov decision process [31], which is defined by a tuple $(\mathcal{S}, \mathcal{A}, R, P, \rho_0, \gamma)$ of the state space $\mathcal{S}$, action space $\mathcal{A}$, reward function $R(\mathbf{s}, \mathbf{a})$, transition probability $P(\mathbf{s}'|\mathbf{s}, \mathbf{a})$, initial state distribution ρ_0, and discounting factor γ. We define a policy $\pi(\mathbf{a}|\mathbf{s})$ that maps from a state $\mathbf{s}$ to an action $\mathbf{a}$. Our objective is to learn the optimal policy that maximizes the expected discounted return, $\mathbb{E}_{\mathbf{s}_0 \sim \rho_0, (\mathbf{s}_0, \mathbf{a}_0, \ldots, \mathbf{s}_{T_i}) \sim \pi} \left[\sum_{t=0}^{T_i - 1} \gamma^t R(\mathbf{s}_t, \mathbf{a}_t) \right]$, where T_i is the variable episode length.

Unlabeled Offline Data: In addition, we assume access to a reward-free task-agnostic dataset [16, 12], which is a set of N state-action trajectories, $\mathcal{D} = \{\tau_1, \ldots, \tau_N\}$, of the agent acting in the environment. Since it is task-agnostic, this data can be easily collected from training data for other tasks, autonomous exploration, or human teleoperation. We do not assume this dataset contains solutions for the downstream task. Therefore, tackling the downstream task requires re-composition of skills learned from diverse trajectories.

Skill-based RL: In this paper, we define skills as a fixed sequence of primitive actions $\{\mathbf{a}_t, \ldots, \mathbf{a}_{t+H-1}\}$ and parameterize skills as a skill latent $\mathbf{z}$ and skill policy, $\pi^L(\mathbf{a}|\mathbf{s}, \mathbf{z})$, that maps a skill latent to the action sequence. Each skill has a fixed length H, following prior works [32, 12, 17].

3.2 Skill-based Model-based Reinforcement Learning (SkiMo)

SkiMo consists of two phases: (1) extract skills and a skill dynamics model from a task-agnostic dataset, and (2) learn a high level policy that acts in the extracted skill space to solve the downstream task, while leveraging the skill dynamics model.

Unlike prior works that learn models completely online [3, 33], we leverage existing offline task-agnostic datasets to pre-train a model and skill policy (see Section 3.2.2 for details). This offers the benefit that the model and skills are agnostic to specific reward functions so that they may be used in multiple tasks. Afterwards at the downstream RL phase, the agent continues to finetune the skill dynamics model and learns a high-level policy for the target task (see Section 3.2.3).

3.2.1 SkiMo Model Components

SkiMo consists of three major model components: the skill policy (π^L_θ), skill dynamics model (D_ψ), and task policy (π_ϕ), along with multiple auxiliary components that are trained or finetuned in different phases. For convenience, we label the trainable parameters ψ, θ, ϕ of each component according to which phase they are trained during.

1. **Learned from task-agnostic data and fine-tuned in downstream RL** (ψ): The state representation module (E_ψ) and the skill dynamics model (D_ψ) are fine-tuned in downstream RL to account for new states and dynamics.
2. **Learned only from task-agnostic data** (θ): In addition to E_ψ and D_ψ, we train the observation reconstruction module (O_θ), skill encoder (q_θ), skill prior (p_θ), and skill policy (π^L_θ) from the task-agnostic data.
3. **Learned in downstream RL** (ϕ): We learn a value (Q_ϕ) and a reward (R_ϕ) function, and the high-level task policy (π_ϕ) in the environment.

The following is a summary of the notations of our major model components for reference, whose relationships are illustrated in Figure 2:

$$\begin{aligned}
&\text{Observation:} && \hat{\mathbf{s}}_t = O_\theta(\mathbf{h}_t) \\
&\text{Skill prior:} && \hat{\mathbf{z}}_t \sim p_\theta(\mathbf{h}_t) \\
&\text{Skill encoder:} && \mathbf{z}_t \sim q_\theta((\mathbf{s}, \mathbf{a})_{t:t+H-1}) \\
&\text{Skill policy:} && \hat{\mathbf{a}}_t = \pi^L_\theta(\mathbf{h}_t, \mathbf{z}_t)
\end{aligned}
\qquad
\begin{aligned}
&\text{Representation:} && \mathbf{h}_t = E_\psi(\mathbf{s}_t) \\
&\text{Skill dynamics:} && \mathbf{h}_{t+H} = D_\psi(\mathbf{h}_t, \mathbf{z}_t) \\
&\text{Task policy:} && \hat{\mathbf{z}}_t \sim \pi_\phi(\mathbf{h}_t) \\
&\text{Reward:} && \hat{r}_t = R_\phi(\mathbf{h}_t, \mathbf{z}_t) \\
&\text{Value:} && \hat{v}_t = Q_\phi(\mathbf{h}_t, \mathbf{z}_t)
\end{aligned}
\tag{1}$$

3.2.2 Learning Skill Dynamics Model and Skills from Task-agnostic Data

The goal of pre-training is to extract skills and skill dynamics, as well as learning a skill prior which will be used to guide exploration, from offline data (summarized in Algorithm 1). Specifically, we would like to jointly learn a skill policy and skill dynamics model, instead of learning them separately as in prior work [28, 5, 29]. The key insight is that this joint training could shape the latent skill space $\mathbb{Z}$ in that the skill dynamics model can easily predict the future. We show the effectiveness of joint optimization in Section 4.4. Furthermore, to permit high-dimensional state spaces, we learn a state representation $\mathbf{h_t} = E_\psi(\mathbf{s}_t)$ where E_ψ is the encoding network.

To learn a low-dimensional skill latent space $\mathbb{Z}$ that effectively encodes action sequences, we train a variational auto-encoder (VAE) [34] on our task-agnostic dataset that learns a mapping from state to action sequence through a skill latent. Here the skill encoder q_θ predicts a skill $\mathbf{z}$ and the decoder or low-level policy π^L_θ reconstructs the original action sequence conditioned on $\mathbf{z}$. Thus, once trained, the latent space $\mathbb{Z}$ along with the low level policy π^L_θ gives us a skill latent and skill policy. This gives rise to the first part of our objective function, which is the VAE objective on reconstructing

low-level actions:

$$\mathcal{L}_{\text{VAE}} = -\mathbb{E}_{(\mathbf{s},\mathbf{a})_{t:t+H-1}\sim\mathbf{D}}\Bigg[\underbrace{\sum_{i=0}^{H-1}\log\pi_\theta^L(\mathbf{a}_{t+i}|\mathbf{h}_{t+i},\mathbf{z})}_{\text{Behavioral cloning}} - \beta\cdot\underbrace{KL\big(q_\theta(\mathbf{z}|(\mathbf{s},\mathbf{a})_{t:t+H-1})\parallel p(\mathbf{z})\big)}_{\text{Embedding regularization}}\Bigg], \tag{2}$$

where $\mathbf{z}$ is sampled from q_θ and β is a weighting factor for regularizing the skill latent z distribution to a prior of a $\tanh$ transformed unit Gaussian distribution, $Z \sim \tanh(\mathcal{N}(0,1))$, i.e., minimizing the KL divergence between $q_\theta(\mathbf{z}|\mathbf{s}_{1:H},\mathbf{a}_{1:H})$ and $p(\mathbf{z})$.

To train the skill dynamics model and make sure the skill space is suited for long-term prediction, we jointly train the skill VAE with a skill dynamics model. The dynamics model is trained to predict the latent state representation H steps ahead (latent state consistency). To train the state representation encoder E_ψ, we also train a state decoder O_θ using reconstruction loss (observation reconstruction). Altogether, this model learns to compactly encode rich information from observations and accurately predict future states with the following objective:

$$\mathcal{L}_{\text{REC}} = \mathbb{E}_{(\mathbf{s},\mathbf{a})_{0:NH}\sim\mathbf{D}}\Bigg[\sum_{i=0}^{N-1}\Big[\underbrace{\lambda_1\|\mathbf{s}_{iH} - O_\theta(E_\psi(\mathbf{s}_{iH}))\|_2^2}_{\text{Observation reconstruction}} + \underbrace{\lambda_2\|D_\psi(\mathbf{h}_{iH},\mathbf{z}_{iH}) - E_{\psi^-}(\mathbf{s}_{(i+1)H})\|_2^2}_{\text{Latent state consistency}}\Big]\Bigg], \tag{3}$$

where $\mathbf{h}_0 = E(\mathbf{s}_0)$ and $\mathbf{h}_{(i+1)H} = D(\mathbf{h}_{iH},\mathbf{z}_{iH})$ such that gradients from the loss are back-propagated through time.

Furthermore, to guide the exploration for downstream RL, we also extract a skill prior (as in [12]) from offline data that predicts the skill distribution for any latent state. The skill prior is trained by minimizing the KL divergence between the skill encoder's output distribution and the skill prior:

$$\mathcal{L}_{\text{SP}} = -\mathbb{E}_{(\mathbf{s},\mathbf{a})_{t:t+H-1}\sim\mathbf{D}}\Big[KL\big(\text{sg}(q_\theta(\mathbf{z}_t|\mathbf{s}_{t:t+H-1},\mathbf{a}_{t:t+H-1}))\parallel p_\theta(\mathbf{z}_t|\mathbf{h}_t)\big)\Big], \tag{4}$$

Combining the objectives above, we train the policy, model, and prior jointly, which leads to a well-shaped skill latent space:

$$\mathcal{L} = \mathcal{L}_{\text{VAE}} + \mathcal{L}_{\text{REC}} + \mathcal{L}_{\text{SP}} \tag{5}$$

3.2.3 Downstream Task Learning with Learned Skill Dynamics Model

For the downstream task, we learn a high-level policy π_ϕ that acts in the skill space with a Dreamer-style model-based RL algorithm. That is, we use the learned skill dynamics model to produce long-horizon imaginary rollouts for efficient planning and downstream learning. Algorithm 2 describes our downstream task training procedure.

To evaluate the imaginary rollouts, we use task-specific rewards to learn a reward function, R_ϕ through reward prediction regression, and a value function, Q_ϕ using fitted Q-iteration following previous work on actor-critic algorithms [35, 36]. We finetune our dynamics model D_ψ and state encoder E_ψ on the downstream task to improve the accuracy of model prediction by optimizing latent state consistency:

$$\begin{aligned}\mathcal{L}'_{REC}(E_\psi, D_\psi, R_\phi, Q_\phi) = \mathbb{E}_{\mathbf{s}_t,\mathbf{z}_t,\mathbf{s}_{t+H}\sim\mathbf{D}}\Bigg[&\underbrace{\|D_\psi(\mathbf{h}_t,\mathbf{z}_t) - E_{\psi^-}(\mathbf{s}_{t+H})\|_2^2}_{\text{Latent state consistency}} + \underbrace{\|r_t - R_\phi(\mathbf{h}_t,\mathbf{z}_t)\|_2^2}_{\text{Reward prediction}}\\ &+ \underbrace{\|r_t + \gamma Q_{\phi^-}(\mathbf{h}_{t+H},\pi_\phi(\mathbf{h}_{t+H})) - Q_\phi(\mathbf{h}_t,\mathbf{z}_t)\|_2^2}_{\text{Value prediction}}\Bigg],\end{aligned} \tag{6}$$

Meanwhile, we train a high-level policy π_ϕ that solves the target task in the skill space . In addition to maximizing the estimated Q-value, we have an auxiliary KL loss that regularizes the high level policy to the learned skill prior p_θ from the pre-training stage (as in [12]). The intuition is that the skill prior learned from task-agnostic data has diverse behaviors that may guide exploration:

$$\mathcal{L}_{RL}(\pi_\phi) = \mathbb{E}_{\mathbf{s}_t \sim \mathbf{D}}\big[-Q_\phi(\mathbf{h}_t, \pi_\phi(\mathbf{h}_t)) + \alpha \cdot KL\big(\pi_\theta(\mathbf{z}_t|\text{sg}(\mathbf{h}_t)) \parallel p_\phi(\mathbf{z}_t|\mathbf{h}_t)\big)\big], \tag{7}$$

To evaluate, we perform MPC with a plan found using CEM following prior work [4]. The key idea is to imagine long-horizon rollouts using the skill policy and skill dynamics model, and choose the best plan according to value and reward predictions. We defer the details of MPC to Appendix, Section A.

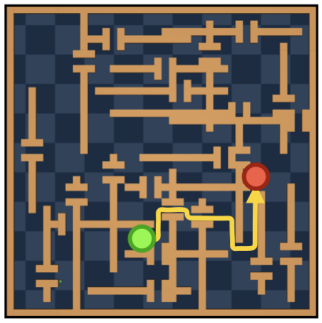
4 Experiments

In this paper, we propose a model-based RL approach that can efficiently and accurately plan long-term future trajectories over the skill space, rather than the primitive action space, by leveraging the skills and skill dynamics model learned from an offline task-agnostic dataset. Hence, in our experiments, we aim to answer the following questions: (1) Can the use of the skill dynamics model improve the efficiency of RL training for long-horizon tasks? (2) Is the joint training of skills and skill dynamics model essential for efficient model-based learning? (3) How does our method's efficiency and performance compare against prior work in skill-based RL and model-based RL?

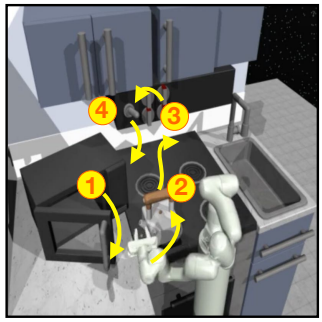
4.1 Tasks

To evaluate whether our method can efficiently learn long-horizon tasks with sparse rewards, we compare it to prior model-based RL and skill-based RL approaches on a 2D maze navigation task and a robotic kitchen manipulation task, as illustrated in Figure 3.

- **Maze** [37, 17]: We use the maze navigation task proposed in Pertsch et al. [17], where a point agent is initialized randomly near the green region and must reach a fixed goal position in the red circle (see Figure 3a). The agent has a 4-dimensional observation space (2-dimensional position, 2-dimensional velocity) and 2-dimensional action space (2-dimensional velocity control). The agent receives a sparse reward of 100 only when it reaches the goal. The task-agnostic offline dataset consists of 3046 trajectories between randomly sampled initial and goal positions.
- **FrankaKitchen** [37, 12, 17]: We use the kitchen manipulation task from the D4RL benchmark [37]. The 7-DoF Franka Panda robot arm needs to perform four sequential tasks (open microwave, move kettle, turn on bottom burner, and flip light switch). The agent has a 30-dimensional observation space (11-dimensional robot proprioceptive state, 19-dimensional object states) and 9-dimensional action space (7-dimensional joint velocity, 2-dimensional gripper velocity). The agent receives a reward of 1 for every subtask completion. We use 603 trajectories collected by teleoperation from Gupta et al. [32] as the task-agnostic data.



(a) Maze



(b) FrankaKitchen

Figure 3: (a) The agent (green) navigates the maze to reach the goal (red). (b) The robotic arm must complete four predefined sub-tasks in the correct order (Microwave-Kettle-Bottom Burner-Light switch).

4.2 Baselines

We compare our method to the state-of-the-art model-based RL method (Dreamer [3]), skill-based RL method (SPiRL [12]), and a combination of them, as well as two ablations of our method.

- Dreamer [3]: Learns a single-step dynamics model and selects actions by latent imagination to achieve a high sample efficiency.
- SPiRL [12]: Trains a skill prior to guide the high-level skill policy for long-horizon learning.
- SPiRL + Dreamer: Learns a policy and model using Dreamer; but instead of acting over the action space, act over the skill space previously extracted from offline data by SPiRL. Unlike SkiMo, this baseline does not train the model and skills jointly or use CEM to plan.

Table 1: Comparison to prior work and ablated methods.

Method	Skill-based	Model-based	Joint training
Dreamer [3]	✗	✓	✗
SPiRL [12]	✓	✗	✗
SPiRL + Dreamer	✓	✓	✗
SkiMo + SAC	✓	✗	✗
SkiMo w/o joint training	✓	✓	✗
SkiMo (Ours)	✓	✓	✓

- SkiMo w/o joint training: The gradient of the low-level skill policy does not flow to the high-level model and vice versa.
- SkiMo + SAC: Chooses skills purely based on the learned high-level policy. This is a model-free variant of our method and it tests the benefit of using the skill dynamics model to plan.

Code for the baselines and environments are included in the supplementary material, and the details of implementation can be found in Appendix, Section A.

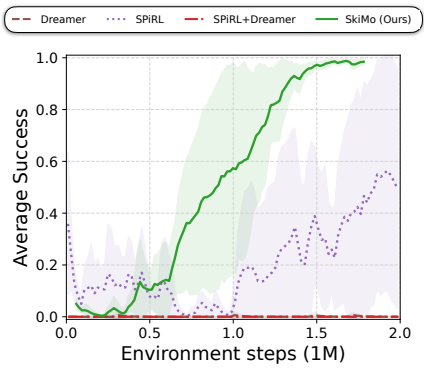
4.3 Results

SkiMo achieves at least 5x higher sample efficiency than all baselines on FrankaKitchen, and is the only method that consistently solves the long-horizon maze navigation.

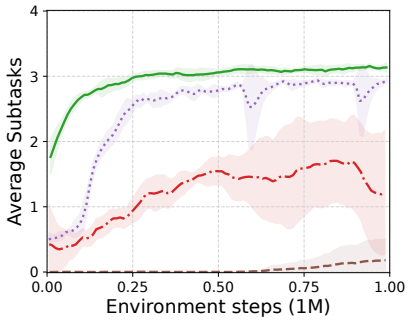
Maze We evaluate SkiMo on the maze navigation task. This task poses a hard exploration problem due to the sparsity of the reward: the agent only receives a binary reward signal after taking 1000+ steps to reach the goal. We hypothesize that learned skills are critical for long-horizon exploration, and a learned model can help the agent summarize its prior experience when external feedback is sparse. To stress-test the agent's ability to leverage reusable skills and the skill model, we also randomize the start location.

In Figure 4 (a) we show that only SkiMo is able to consistently succeed in long-horizon navigation. Prior methods struggle to learn a goal-reaching policy or an accurate model due to the challenge in long-horizon learning and the sparsity of feedback. To better understand the result, we qualitatively analyze the behavior of each agent in Appendix, Section B. Based on the visualization, Dreamer has a small coverage of the maze, since it is challenging to coherently explore for 1000+ steps to reach the goal from taking primitive actions. SPiRL is able to explore the most fraction of the maze, but it does not consistently find the goal. SPiRL + Dreamer, despite learning a policy in the skill space, never reaches the goal since it often selects skills that lead to collisions; its model, which learns separately from skills, receives no learning signal when the agent does not get the reward and thus it has to make random predictions.

FrankaKitchen We then investigate SkiMo's ability to learn complex robotic manipulation tasks in the kitchen environment. It tests the robot agent's ability to compose skills such as pulling, picking, placing, turning, and flipping objects and execute such skills in a correct sequence.



(a) Maze



(b) FrankaKitchen

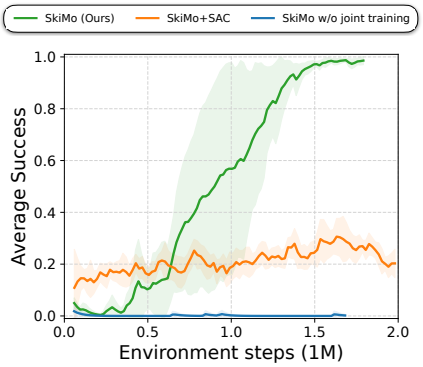
Figure 4: **Learning curves of our method and baselines** (each averaged over 5 random seeds).

Prior work, SPiRL [12], relegates the success on kitchen to the temporal abstraction of primitive actions. We show in Figure 4 (b) that SkiMo is able to reach the same performance with 5x less environment interactions than SPiRL. This improvement in sample efficiency is crucial in the real world robot learning. We attribute our high sample efficiency to the use of the skill dynamics model. On the other hand, Dreamer rarely successfully manipulates objects due to the difficulty in long-horizon learning with primitive actions. SPiRL + Dreamer performs better than Dreamer by leveraging existing skills, but its model and policy, which are learned separately from skill training, are not strong enough to consistently achieve more than two sub-tasks. We further ablate our model to investigate the major contributors in the next section.

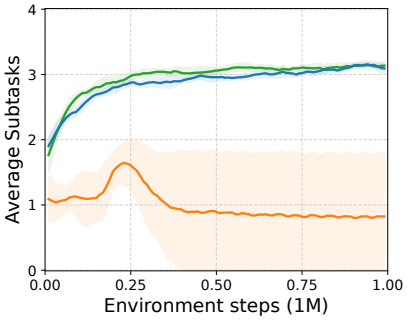
4.4 Ablation Studies

Can the use of the skill dynamics model improve the efficiency of RL training for long-horizon tasks? Comparing to SkiMo + SAC, which directly uses the high-level policy to select skill embeddings instead of using the skill dynamics model to plan, SkiMo achieves better asymptotic performance and higher sample efficiency on both maze navigation and robotic manipulation tasks (Figure 5). Since the only difference is in the use of a skill dynamics model, this comparison suggests that the high-level actor can make more informative decisions on long-horizon tasks by leveraging the skill dynamics model, which can be trained to make accurate long-term predictions.

Is the joint training of skills and skill dynamics model essential for efficient model-based learning? On top of the skill dynamics model, we find that jointly training it with skills is critical for tasks with very long horizons where the model needs to make accurate long-term predictions. Comparing to SkiMo, SkiMo w/o joint training does not learn on the maze navigation task, since the latent space is shaped only for skill reconstruction, not long-term predictions. On the other hand, we also find that joint training only makes marginal contribution to tasks like kitchen where long-term prediction is no longer the bottleneck. Overall, we still find the jointly training to be beneficial for model learning. For further ablations and discussion, see Appendix, Section C.



(a) Maze



(b) FrankaKitchen

Figure 5: **Learning curves of our method and ablated models** (each averaged over 5 random seeds).

5 Conclusion

We propose an intuitive instantiation of saltatory model-based hierarchical RL [2] that combines skill-based and model-based RL approaches. Our experiments demonstrate that (1) a skill dynamics model can improve the performance of prior model-based RL and skill-based RL, and (2) joint training of the skill dynamics and skill representations further improves the sample efficiency by learning skills useful to predict their consequences. As a future work, we would like to apply this sample-efficient approach to real robots where the sample efficiency is crucial. We believe that the ability to learn and utilize a skill-level model holds the key to unlocking the sample efficiency and widespread use of RL agents, and our method takes a step toward this direction.

Limitations While our proposed method extracts fixed-length skills from offline data, the lengths of semantic skills in the real world vary based on the context and goals. Future work can learn variable-length semantic skills to improve long-term future prediction and planning. Further, even though we only experimented on state-based inputs, SkiMo is a general framework that can be extended to RGB, depth, or tactile-based observations to improve the sample efficiency of robot learning in the real world.

References

[1] S. Legg and M. Hutter. Universal intelligence: A definition of machine intelligence. *Minds and machines*, 17(4):391–444, 2007.

[2] M. Botvinick and A. Weinstein. Model-based hierarchical reinforcement learning and human action control. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369 (1655), 2014.

[3] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2019.

[4] N. Hansen, X. Wang, and H. Su. Temporal difference learning for model predictive control. In *International Conference on Machine Learning*, 2022.

[5] K. Lu, A. Grover, P. Abbeel, and I. Mordatch. Reset-free lifelong learning with skill-space planning. In *International Conference on Learning Representations*, 2021.

[6] K. Lowrey, A. Rajeswaran, S. Kakade, E. Todorov, and I. Mordatch. Plan online, learn offline: Efficient learning and exploration via model-based control. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Byey7n05FQ.

[7] M. Janner, J. Fu, M. Zhang, and S. Levine. When to trust your model: Model-based policy optimization. In *Neural Information Processing Systems*, 2019.

[8] A. Argenson and G. Dulac-Arnold. Model-based offline planning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=OMNB1G5xzd4.

[9] R. S. Sutton, D. Precup, and S. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[10] Y. Lee, S.-H. Sun, S. Somasundaram, E. S. Hu, and J. J. Lim. Composing complex skills by learning transition policies. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rygrBhC5tQ.

[11] Y. Lee, J. Yang, and J. J. Lim. Learning to coordinate manipulation skills via skill behavior diversification. In *International Conference on Learning Representations*, 2020.

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

[13] Y. Lee, J. J. Lim, A. Anandkumar, and Y. Zhu. Adversarial skill chaining for long-horizon robot manipulation via terminal state regularization. In *Conference on Robot Learning*, 2021.

[14] M. Dalal, D. Pathak, and R. Salakhutdinov. Accelerating robotic reinforcement learning via parameterized action primitives. In *Neural Information Processing Systems*, 2021.

[15] Y. Lee, E. S. Hu, and J. J. Lim. IKEA furniture assembly environment for long-horizon complex manipulation tasks. In *IEEE International Conference on Robotics and Automation*, 2021. URL https://clvrai.com/furniture.

[16] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet. Learning latent plans from play. In *Conference on Robot Learning*, pages 1113–1132. PMLR, 2020.

[17] K. Pertsch, Y. Lee, Y. Wu, and J. J. Lim. Demonstration-guided reinforcement learning with learned skills. In *Conference on Robot Learning*, 2021.

[18] D. Ha and J. Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[19] R. Mendonca, O. Rybkin, K. Daniilidis, D. Hafner, and D. Pathak. Discovering and achieving goals via world models. In *Neural Information Processing Systems*, 2021.

[20] R. Y. Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112, 1997.

[21] G. Williams, A. Aldrich, and E. Theodorou. Model predictive path integral control using covariance variable importance sampling. *arXiv preprint arXiv:1509.01149*, 2015.

[22] T. Wang and J. Ba. Exploring model-based planning with policy networks. *arXiv preprint arXiv:1906.08649*, 2019.

[23] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou. Aggressive driving with model predictive path integral control. In *IEEE International Conference on Robotics and Automation*, page 1433–1440, 2016.

[24] A. Sharma, S. Gu, S. Levine, V. Kumar, and K. Hausman. Dynamics-aware unsupervised discovery of skills. In *International Conference on Learning Representations*, 2020.

[25] R. S. Sutton, D. Precup, and S. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[26] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal. Learning and generalization of motor skills by learning from demonstration. In *IEEE International Conference on Robotics and Automation*, pages 763–768, 2009.

[27] O. Nachum, H. Tang, X. Lu, S. Gu, H. Lee, and S. Levine. Why does hierarchy (sometimes) work so well in reinforcement learning? *arXiv preprint arXiv:1909.10618*, 2019.

[28] B. Wu, S. Nair, L. Fei-Fei, and C. Finn. Example-driven model-based reinforcement learning for solving long-horizon visuomotor tasks. In *Conference on Robot Learning*, 2021.

[29] K. Xie, H. Bharadhwaj, D. Hafner, A. Garg, and F. Shkurti. Latent skill planning for exploration and transfer. In *International Conference on Learning Representations*, 2020.

[30] D. Shah, A. T. Toshev, S. Levine, and brian ichter. Value function spaces: Skill-centric state abstractions for long-horizon reasoning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=vgqS1vkkCbE.

[31] R. S. Sutton. *Temporal credit assignment in reinforcement learning*. PhD thesis, University of Massachusetts Amherst, 1984.

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

[33] R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak. Planning to explore via self-supervised world models. In *ICML*, 2020.

[34] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[35] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[36] 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*, pages 1856–1865, 2018.

[37] 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.

[38] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

[39] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. de Las Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, T. P. Lillicrap, and M. A. Riedmiller. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[40] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems*, pages 305–313, 1989.

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