SKILL GRAPH FOR REAL-WORLD QUADRUPEDAL ROBOT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep Reinforcement Learning (DRL) is one of the promising methods for general learning policies from the environment. However, DRL has two basic problems: sample inefficiency and weak generalization. Real-world robotic DRL, for example, often requires time-consuming data collection and frequent human intervention to reset the environment. If faced with a new environment or task, the robot can master basic skills in advance instead of learning from scratch, then its learning efficiency and adaptability will be greatly improved. Therefore, in this paper, we propose a novel structured skill graph (SG) for accelerating the learning of robotic DRL policies and rapid adaptation to unseen real-world tasks. Similar to the knowledge graph (KG), SG adopts the tri-element structure to store information. But different from KG storing static knowledge, SG can store dynamic policies and adopt different tri-elements. To construct the SG, we utilize various real-world quadrupedal locomotion skills in different realistic environments. When faced with new real-world tasks, the relevant skills in SK will be extracted and used to help the robotic DRL learning and rapid adaptation. Extensive experimental results on the real-world quadruped robot locomotion tasks demonstrate the effectiveness of SG for facilitating DRL-based robot learning. Real-world quadrupedal robots can adapt to new environments or tasks in minutes with the help of our SG.

1 INTRODUCTION

How to efficiently combine human knowledge in intelligent systems is a typical research direction of artificial intelligence. Inspired by human problem solving, knowledge of presentation and reasoning are key for intelligent systems to solve challenging tasks (Shortliffe, 2012). Recently, knowledge graph (KG), as a structured storage form of human knowledge, have attracted great attention from academia and industry (Hogan et al., 2020; Ji et al., 2022; Chaudhri et al., 2022). The KG is a knowledge base of information about entities that uses a collection of subject-predicate-object triples (also known as facts) to represent entities and their relations. In the KG, nodes represent entities, and edges between nodes reflect the relations between entities. Nowadays, the KG has been widely used in the recommendation, question answering, text generation, and other fields (Lehmann et al., 2015; Li et al., 2020; Erxleben et al., 2014; Mahdisoltani et al., 2014). However, existing KG usually focuses on text processing and pays little attention to the various dynamic behavioral or skill information possessed by agents or robots.

To acquire any dynamic behaviors and skills of agents, deep reinforcement learning (DRL) (Sutton & Barto, 2018) is a general and powerful learning framework. In recent years, it has made some major breakthroughs in the fields of game environments (Silver et al., 2016; Berner et al., 2019; Vinyals et al., 2019), robotic manipulation behaviors (Kalashnikov et al., 2018; Ebert et al., 2018), quadrupedal locomotion (Hwangbo et al., 2019; Lee et al., 2020; Miki et al., 2022) and so on. Unfortunately, DRL has suffered from two fundamental problems: sample inefficiency and poor generalization performance (Kirk et al., 2021). To alleviate these issues, many works try to leverage meta RL (Rakelly et al., 2019; Li et al., 2021; Pong et al., 2022; Yuan & Lu, 2022; Lin et al., 2022; Wang & van Hoof, 2022), skill-based RL (Pertsch et al., 2020; Nam et al., 2022; Shankar & Gupta, 2020; Shankar et al., 2022), multi-task RL (Yang et al., 2020b; D'Eramo et al., 2020; Zhang & Wang, 2021; Sodhani et al., 2021; Yu et al., 2021; Hong et al., 2022) and other methods (Xu et al., 2022).

Most of these works are still limited to simple simulation environments and struggle to perform well in challenging real-world tasks.

The fundamental problems of DRL methods also exist for real-world DRL-based quadrupedal locomotion research. Although several recent works have achieved outstanding breakthroughs (Hwangbo et al., 2019; Lee et al., 2020; Yang et al., 2020a; Miki et al., 2022), the stable control is a challenging point for DRL-based methods. The action space of the quadrupedal robot is a high-dimensional (12 and above) continuous space. It is difficult for learning-based methods to find the optimal solution in a high-dimensional continuous space from scratch.

To perform well on real-world quadrupedal locomotion tasks, many works introduce prior knowledge during the training of DRL policies. The prior knowledge greatly improves the training efficiency and generalization ability of policies. It is usually represented in a variety of forms, such as ideal motion data (Peng et al., 2020; Vollenweider et al., 2022), trajectory generators (Iscen et al., 2018; Rahme et al., 2020), evolutionary trajectory generator (Thor et al., 2021; Shi et al., 2022a), control methods (Yang et al., 2021; Gangapurwala et al., 2021), and so on. However, in most of these studies, prior knowledge only plays an auxiliary role in policy learning. The specific construction of prior knowledge is somewhat arbitrary and requires manual parameter tuning. For example, in the design of trajectory generators, the swing trajectory of the leg is generally considered to have a certain periodicity. But generators can only be designed for specific tasks. Faced with new tasks and environments, we need to spend time redesigning the form of the generator. As a result, such generators are unstructured and difficult to extend. Borrowing ideas from KG construction, our work aims to build structured prior knowledge for quadruped robots. The structuring of prior knowledge means that robots can learn and adapt quickly when faced with new tasks or environments with the help of prior knowledge.

In this paper, we propose a novel skill graph (SG) for real-world quadrupedal robots to enable fast learning of DRL policies. Compared with the common KG, our proposed SG is mainly aimed at the dynamic behavior and skills of the realistic robot. Moreover, our robotic behavior data is structured, thus the construction of SG focuses on the definition and representation of entities, attributes and relations. Another feature is that the constructed SG is highly scalable and will be widely utilized for rapid learning, transfer and generalization of real-world robotics tasks. Specifically, we first formulate various real-world quadruped locomotion tasks and collect a large amount of behavioral data. The structure behavioral data are then utilized to construct the robotic SG, to complete the representation of skills. Next, according to the downstream real-world tasks, relevant skills are extracted from the SG. The highly relevant skills can help robots adapt to new challenging real-world tasks. The main contributions of our paper are as follows:

- We construct a novel realistic robotic SG containing 2 quadrupedal robots, 5 common environments, and 844 quadrupedal robot skills. The SG can structure prior knowledge in real-world DRL-based quadrupedal locomotion research.
- The robotic SG can be utilized to visualize skills, thereby facilitating the development and maintenance of skills. It can also greatly facilitate fast learning of DRL policies in the face of downstream real-world tasks.
- Experiments on realistic robots demonstrate the effectiveness of our proposed SG. The quadrupedal robot can acquire novel skills and adapt to new environments in minutes.

2 Related Work

2.1 Integrating commonsense knowledge into DRL agents

There has been some recent work on how to integrate commonsense knowledge into DRL agents. (Jiang et al., 2020) built a commonsense DRL simulation environment and used information from external KG to guide the learning of DRL agents. (Murugesan et al., 2021) designed a text-based game environment for training and evaluating RL agents with commonsense knowledge. They also introduced several baseline DRL agents that track sequential context and dynamically retrieve relevant commonsense knowledge from ConceptNet. (Höpner et al., 2022) utilized subclass relations in open source knowledge graphs to abstract specific objects and developed a residual policy gradient method that integrates knowledge across different abstraction levels in class hierarchies. (Ammanabrolu & Riedl, 2021) proposed a KG-based world model, a multi-task transformer-based architecture that learns to simultaneously generate a set of graph disparities and a set of context-dependent actions. (Zhao et al., 2022) proposed a dynamic knowledge and skill graph (KSG) and developed a specific KSG based on CNDBpedia. The KSG can search for the skills of different agents in different environments, providing transferable information for acquiring new skills. While these works are limited to simple simulation environments or text-based games, the SG we build is based on realistic quadrupedal locomotion data.

2.2 Skill-based RL

In common skill-based RL, skills are generally represented as sub-policies or a series of low-level actions to facilitate the learning of long-horizon behaviors. Many works propose having the agent take action on time-expanding skills, such as options (Sutton et al., 1999; Shankar & Gupta, 2020; Shankar et al., 2022) or motion primitives (Pastor et al., 2009; Pertsch et al., 2020; Salter et al., 2022; Rao et al., 2022; Pertsch et al., 2021). Intuitively, temporal abstraction can effectively reduce the task horizon of the agent and enable directed exploration, which is a major challenge for DRL agents facing challenging tasks (Nachum et al., 2019). However, skill-based RL struggles with real-world tasks and requires a large number of environment interactions (Lee et al., 2021). (Shi et al., 2022b) used model-based RL to guide skill planning to improve the sample efficiency of skill-based approaches. In contrast to these works, we build structured realistic skills with SG, enabling DRL agents efficiently adapt to complex real-world tasks.

2.3 Offline Meta-RL

While skill-based RL generally requires high-quality offline data (that is, data collected by expert policies), offline meta RL does not have this hard requirement. Instead, such methods require (sub-optimal) offline data containing reward functions or task annotations (Nam et al., 2022; Mitchell et al., 2021; Dorfman & Tamar, 2020; Dorfman et al., 2020; Pong et al., 2022; Shi et al., 2022b). These works first meta-train DRL agents using pre-collected offline datasets. Then, they aim to rapidly adapt the agent to unseen tasks, assuming only limited access to data from new tasks. These methods usually require the offline training data to be divided into separate datasets for each training task. The task distribution is compact and the difference among tasks is small. So these methods struggle to generalize to more different tasks. Unlike these works, the SG we build will help agents quickly adapt to new and more difficult real-world tasks.

2.4 Prior knowledge in real-world DRL-based quadrupedal locomotion

In the realistic DRL-based quadrupedal locomotion research, prior knowledge is represented in a variety of forms, such as motion data (Singla et al., 2019; Peng et al., 2020; Vollenweider et al., 2022; Bohez et al., 2022), trajectory generators (Iscen et al., 2018; Jain et al., 2019; Rahme et al., 2020; Zhang et al., 2021), control methods (Yang et al., 2021; Gangapurwala et al., 2021; Yao et al., 2021), and so on. Motion data is often generated by other sub-optimal controllers or public datasets. Through imitation learning or other methods, the robot can obtain natural and agile motion patterns, and then complete specified tasks according to the external reward. Trajectory generators and control methods generally introduce priors into the action space of DRL policies to narrow the search range of actions. This greatly reduces the learning difficulty of the robot and improves their sample efficiency. Compare with these methods, our work aims to construct structured skill priors for studying rapid adaptation and fast learning capabilities in real-world quadrupedal locomotion tasks.

3 Preliminaries

The standard framework of RL is Markov decision processes (MDPs) specified by the tuple $\mathcal{M} := (\mathcal{S}, \mathcal{A}, r, P, \rho_0, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $r(\mathbf{s}, \mathbf{a})$ is the reward function, $P(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ is the stochastic transition dynamics, $\rho_0(\mathbf{s})$ is the initial state distribution, and γ is the discount factor. The goal in RL is to learn a policy $\pi(\mathbf{a}|\mathbf{s})$ that maximizes the expected discounted reward $\eta(\pi) := \mathbb{E}_{\tau\sim p^{\pi}(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t r(\mathbf{s}_t, \mathbf{a}_t)\right]$, where $\tau := (\mathbf{s}_0, \mathbf{a}_0, r_0, \mathbf{s}_1, \mathbf{a}_1, r_1, ...)$ represents a trajectory. The action-value function $Q(\mathbf{s}, \mathbf{a})$ is the discounted return obtained

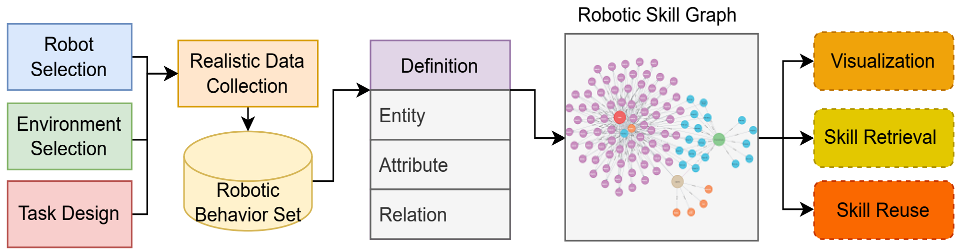


Figure 1: The construction process of our proposed SG. We first identify the realistic quadruped robot and environment, as well as formulate the real-world tasks. Then the empirical data is collected according to each task, and is further trained to obtain the robot's behavior set. Following this, we definite and represent the entities, attributes and relations in the behavior set. The connection among the behaviors is obtained, and the robotic SG is constructed accordingly. The constructed SG is finally leveraged for visualization, skill retrieval and reuse.

by executing action $\mathbf{a}$ at current state $\mathbf{s}$ and then following the policy $\pi(\mathbf{a}|\mathbf{s})$: $Q(\mathbf{s},\mathbf{a}) := \mathbb{E}_{\tau\sim p^{\pi}(\tau)}\left[\sum_{t=0}^{\infty}\gamma^{t}r(\mathbf{s}_t,\mathbf{a}_t)|\mathbf{s}_0=\mathbf{s},\mathbf{a}_0=\mathbf{a}\right]$. A typical actor-critic method alternates between the policy evaluation and policy improvement phases. In policy evaluation, we fitting the action-value function $Q(\mathbf{s},\mathbf{a})$ to evaluate the current policy $\pi(\mathbf{a}|\mathbf{s})$. The policy $\pi(\mathbf{a}|\mathbf{s})$ is then updated to maximize the target Q-value in the policy improvement phase.

4 Construction and Application of Robotic Skill Graph

In this paper, we aim to build a robotic skill graph (SG) that consists entirely of the skills of real-world quadrupedal robots. The SG can provide transferable skills for realistic robots to learn novel skills and adapt to new environments. The construction process of SG is shown in Figure 1, and can be roughly divided into three parts: behavioral data preparation, definition and representation of the SG elements, and SG application. Construction details will be explained further below.

4.1 Data Preparation for Skill Graph

In the data preparation phase, we need to design the environment and tasks of the real-world quadrupedal robot. Specifically, the hardware structure of the robot and the stability of the robot behavior need to be considered. In the initial release of our proposed SG, some simple but necessary environments are included, such as marble flat, marble slope, grass, etc. The tasks of the robot are to track the desired locomotion target. The specific design results in the experiment section.

After the environment and tasks are determined, we utilize realistic quadruped robots to collect a large amount of empirical data. These data are stored in the form of empirical pairs (s_t, a_t, r_t, s_{t+1}). The action a_t is the desired joint angle (12-dimension). The state s_t is a 44-dimensional continuous vector, which contains COM linear velocity (2 dims), attitude angle (3 dims) and angular velocity (3 dims), joint angle (12 dims) and joint angular velocity (12 dims), action at the last time step (12 dims). When designing the reward function r_t, the locomotion target and energy consumption of the robot need to be considered: $r = r_1 + r_2 + r_3 + 0.001 * r_4$, where $r_1 = e^{-\sum(\hat{v}-v)^2/0.025}$, $r_2 = e^{-\sum(\hat{\omega}-\omega)^2/0.025}$, $r_3 = e^{-\sum(\hat{p}-p)^2/0.025}$, and $r_4 = -\sum\tau^2/12$. $\hat{v}, v, \hat{\omega}, \omega, \hat{p}, p$ and τ represent the desired linear velocity, current linear velocity, desired yaw rate, yaw rate, desired pitch angle, pitch angle and desired torque, respectively.

In terms of robotic behaviors design, we find that, compared with various latent variable operations in context-based meta-RL, it is more effective to directly combine the basic skills based on DRL policies to learn to solve challenging tasks (Yang et al., 2020a). Therefore, we leverage an efficient offline RL algorithm CQL (Kumar et al., 2020) to train the policy on the collected empirical data. The trained policy network and value network are utilized as representations of behaviors. Since real-world data collection is quite time-consuming and labor-intensive, the data scale of a single task is not large. Meanwhile, limited by the sensor accuracy of the robot, the data contains a noise of different stochastic degrees. To alleviate these issues, we leverage a simple and efficient data augmentation approach. Inspired by (Sinha et al., 2021), we add a small amount of Gaussian noise to the state s_t of the collected data before policy training.

4.2 The Definition of Entities, Attributes, and Relations

To build the robotic SG, we need to define the available knowledge units, including entities, attributes, and relations. Three types of entity nodes are considered in SG: quadrupedal *robot*, *environment* and *skill*.

Different from common KG, we innovatively introduce dynamic behavioral skill information to construct robotic SG. The *robot*, *environment* and *skill* will act as the entities whose specific attributes need to be defined. The attributes of the *robot* entities are straightforward since different robots have different mechanical structures and dynamic models. So the robots' physical characteristics (such as mass, inertia, body length, and leg length) can be used as attributes of the *robot* entities. For the definition of the attributes of the *skill* entities, since skills are highly related to the task, the robot's desired tracking locomotion target is a reasonable option. However, the definition of the attributes of the *environment* entities is slightly more complicated. In multi-task RL, one-hot encoding is usually used to represent tasks (or environments). it is generally assumed that tasks (or environments) are independent and identically distributed. However, one-hot encoding is too simplistic for real-world robotic tasks, which is not conducive to the rapid adaptation of robots to new tasks (or environments). We utilize physical quantities (friction coefficient, slope, etc.) as a better choice for *environment* entity properties.

An entity relation is an association between entities that specifies how entities are connected. In our proposed robotic SG, we mainly focus on the relations among *robot*, *environment* and *skill* entities. There are two types of entity relations: discrete and continuous. The relations among the three different kinds of entities (*robot*, *environment* and *skill*) are discrete. That is, the relation exists (can) or does not exist (cannot). Moreover, the relations among entities of the same label are continuous, and these relations are established using the similarity metric. Entities with higher similarity are more closely related, and vice versa.

Similar to KG, SG adopts tri-elements $\langle entity, relation, entity \rangle$ structure to store dynamic skill information. For example, for *robot A* and *environment B* entities, the tri-element $\langle robot\ A, in, environment\ B \rangle$ in SG can be expressed as a quadrupedal *robot A* can demonstrate skills in *environment B*. For *skill C* and *skill D* entities, the tri-element $\langle skill\ C, 0.8\ similarity, skill\ D \rangle$ in SG can be expressed as the *skill C* and *skill D* have a similarity of 0.8.

4.3 Applications of Skill Graph

An important application of SG is the visualization of robotic skills. The number and relation of *robot*, *environment* and *skill* entities can be clearly displayed. The SG can also show all the skills of the robot in the same environment. Users can easily understand the relation between these entities, and then better analyze, construct and utilize robotic skills. Different from the introduction of prior knowledge in previous work, our proposed SG is highly structured and easy to extend and maintain.

Our proposed SG can also provide the skill retrieval function for entities, which is mainly divided into two parts. The first part is the retrieval of *robot*, *environment* and *skill* entities, which can be divided into three types: label retrieval, attribute retrieval and entity relation retrieval. The SG defines three entity labels: robot, environment and skill. Attributes of entities with different labels are different. Users can directly query which entities are in the SG according to the label. In the SG, *environment* and *skill* entities have unique attributes, which are environment characteristics and skill parameters, respectively. Users can further use attributes to specify entities and retrieve entity nodes that satisfy specific relations between entities.

The second part is to match related skills based on similarity metrics. When the robot is solving real-world tasks, if the required skills are in the SG, then we directly retrieve these skills. But if the required new skill is not in the SG, we need to first calculate the similarity between the new environment and the existing environment, then select the most similar skill in the most similar environment. The specific skill retrieval process is shown in Algorithm 1.

Algorithm 1 Skill Retrieval Based on Similarity of Entity

Input: Agent name: A, Environment feature: E_f, The desired skill parameters: $desired\_P$.

MATCH (a:Agent{name: A}) RETURN a;
▷ Retrieve the given agent a according to agent name
MATCH (e:Environment{Feature: E_f}) RETURN e;
▷ Retrieve the given environment e according to environment feature
if The environment e is inexistent **then**
  MATCH (a:Agent{name: A}) $\rightarrow$ ($Envs$:Environment); RETURN $Envs$
  ▷ Retrieves all environments $Envs$ associated with a given agent a
  Calculate the feature similarity between e and $Envs$;
  Select the most similar environment E_s, RETURN e = E_s;
end if
MATCH (a:Agent) $\rightarrow$ (e:Environment) $\rightarrow$ (s:Skill) RETURN s;
▷ Retrieve all skills of agent a in Environment e
Calculate the similarity of skill parameters between the desired skill and retrieved skills s;
Select the most similar skill s, RETURN s.

Skills retrieved in SG will be further fine-tuned, so that the robot can quickly learn novel skills and adapt to the new environment. Specifically, the policy and value network retrieved in SG will serve as initial networks. These initial networks are further trained according to the online RL algorithm SAC (Haarnoja et al., 2018). Only a limited number of realistic samples and training time are utilized in this fine-tuning process. The novel skill learned will be added to the robotic SG according to the construction rules to realize the continuous learning of the robot.

5 Experiments

In this section, we aim to validate the functionality of SG and its facilitation for DRL policy learning on real-world tasks in quadruped robots. Firstly, we define several evaluation metrics about real-world tasks. We then illustrate some algorithmic baselines that will be compared with our proposed method. Furthermore, the SG is visually displayed in several typical cases. Finally, in multiple real-world scenarios, we verify the skill retrieval and reuse function of SG, as well as quantitatively analyze its promoting effect on DRL policy learning.

5.1 Experimental Setup

Metrics: For real-world quadrupedal locomotion tasks, we utilize two different types of evaluation metrics: cumulative undiscounted reward (Return), and cost of transportation (COT). we first define Return: $M_1 = \sum_t^T r_t$, where T is the number of real-world interactions. The Return metric is the most important metric for the DRL community, and directly evaluates the robot's performance on new real-world tasks. We also utilize the COT to compare the energy consumption of DRL policies on real-world tasks: $M_2 = \sum_t^T [(|\tau_t \dot{q}_t|)/(mg\|v_t\|_2)]/T$, where mg and v are the total weight and linear velocity of the robot, respectively. COT is a common metric in the legged locomotion research field, since it quantifies the positive mechanical power applied by the actuator per unit weight and unit locomotion speed (Collins et al., 2005).

Baselines: We compare the following baselines: 1) **SAC**: The SAC is a popular online off-policy DRL algorithm and the one we utilize for new skill learning. So it serves as a weak baseline for policy learning. 2) **Fine-Tuning**: Leveraging a more efficient online off-policy REDQ algorithm (Chen et al., 2021), (Smith et al., 2022a) first learn the robot's forward, backward, and fall standing skills in a simulated environment, and then further learn these skills in the real world. 3) **Dreamer**: (Wu et al., 2022) applied the model-based RL algorithm Dreamer (Hafner et al., 2019; 2020) to a quadruped robot, and learned directly online in the real world without any simulator. They trained a quadruped robot to roll, stand and walk from scratch under 1 hour without resetting. 4) **Efficient RL**: Leveraging the more sample-efficient online off-policy RL algorithm DroQ (Hiraoka et al., 2022) and the machine learning framework JAX (Bradbury et al., 2018), (Smith et al., 2022b) can learn the walking locomotion of the quadrupedal robot directly in the real world in just 20 minutes.

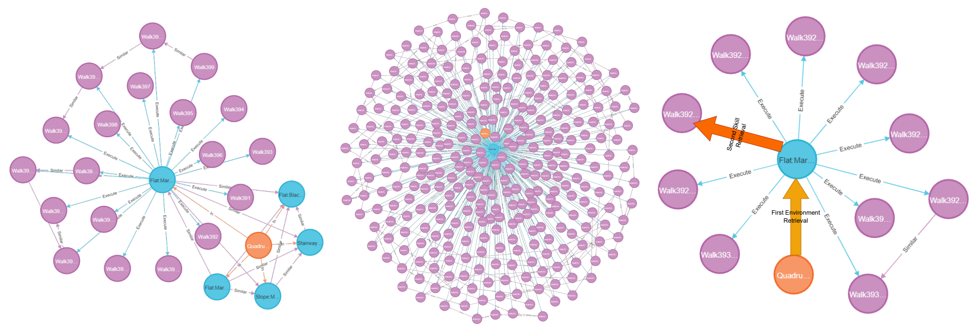


Figure 2: The visualization of our proposed (partial) skill graph. Specifically, we visualize the relation between entities and the relation between environments (Left). Meanwhile, we make a visualization of all the skills of a robot in an environment (Middle). We also visualize the skill retrieval process (Right). Robot, environment and skill entities are represented by orange, blue and purple nodes respectively. The relations between entities are represented by edges between nodes. The connection relation will be displayed only if the similarity between skills is greater than 0.95, which is convenient for visualization. Please refer to Appendix Figure 10, Figure 11, and Figure 12 for the SG's details.

5.2 Task Design Results and Visualization

Task Design Results: For designing real-world quadrupedal locomotion tasks, firstly we need to identify the robot and the environment. In the initial release of SG, we use *Unitree A1*[1] and our own robot as the robot entities of SG, and their attributes are shown in Appendix Figure 7 and Table 3. Then, we set reasonable variables from the aspects of environment and locomotion targets. Five common terrains (indoor ground, outdoor marble plane, etc.) are considered first, as shown in Figure 3. For the design of the attributes of the environment entity, we currently consider the friction coefficient and slope of the ground, as shown in Appendix Table 4.

Figure 3: Five realistic environments considered in the SG.

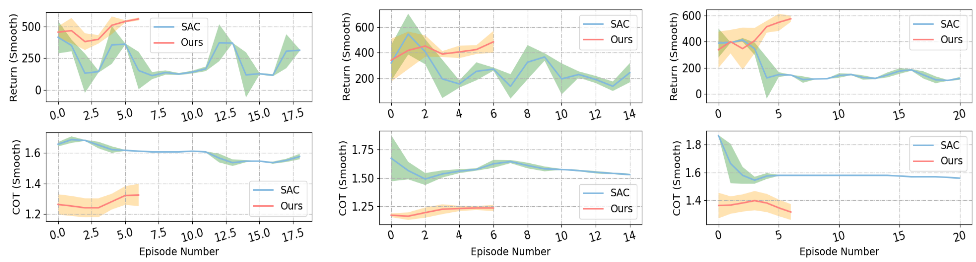
In terms of locomotion target design, we currently mainly set reasonable tracking targets from four variables: $v_x, v_y, d\psi$ and θ. v_x and v_y are the velocities along the x and y axes of the Center of mass (COM) in the world frame. $d\psi$ and θ are the yaw velocity and the pitch angle in the body frame, respectively. We use these four variables to form a vector to represent the locomotion target: $K = (v_x, v_y, d\psi, \theta)$. These continuous variables will be discretized, and the values are shown in Appendix Table 5. The values of the locomotion target vector K are shown in Appendix Table 6. For example, the robot behavior with the locomotion target of $K = (0.1, 0, 0, 0)$ in the indoor environment is the realization of a skill in SG. The initial release of SG contains a total of 844 skills, most of which are collected on the indoor floor, and a small number of skills are collected outdoors, as shown in Table 1.

Visualization: After the SG is constructed, we display it visually, as shown in Figure 2. The left image in Figure 2 mainly shows the relation among the three kinds entities and among the environment entities in the SG. Specifically, the relation between entities can be expressed as: 1) the quadruped robot *Unitree A1* is in an indoor environment; 2) the quadruped robot has a skill whose locomotion target is $K = (0.1, 0, 0, 0)$; 3) a skill whose locomotion target is $K = (0.1, 0, 0, 0)$ can be used in indoor marble ground display. The relation between environments is characterized by similarity. For example, the similarity between indoor floor and outdoor marble floor is higher than that of indoor floor and grass. The middle image in Figure 2 mainly shows the relation between skills in an environment of a robot in SG.

Table 1: The number of skills included in each environment.

Env.	Num.
Indoor Floor	312
Marble Floor	204
Marble Slope	80
Asphalt Road	136
Grassland	112
Total Num.	844

[1]https://www.unitree.com/products/a1/



(a) Task 32 (b) Task 33 (c) Task 34

Figure 4: Return (Upper) and COT (Bottom) of the rapid learning and adaptation process of quadruped robots on new real-world tasks $32, 33$ and 34. Task MN represents the new task number, where M represents the M^{th} environment in Appendix Figure 8 , and N represents the N^{th} locomotion target in Appendix Table 7. The x-axis represents the number of episodes. The shaded area represents one standard deviation. The total sample number for the skills fine-tuning phase is only 5000. Each experiment was repeated three times.

5.3 Skill Retrieval and Reuse

Skill Retrieval: To verify the robot's ability to rapidly learn and adapt when faced with new real-world tasks, we designed six specific tasks. The new environment and locomotion target design details are shown in Appendix Figure 8 and Table 7.

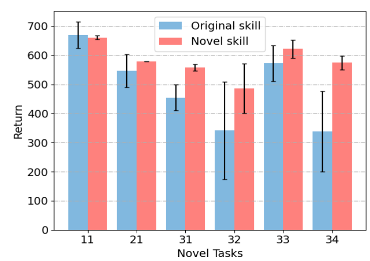


Figure 5: Performance of skills in SG before and after fine-tuning when the robot is faced with six new real-world tasks. The x-axis represents the new task number MN, and the y-axis represents the return score. The solid black line represents one standard deviation. Each experiment was repeated three times.

To realize the rapid learning and adaptation of the robot, we need to perform skill retrieval on the SG. The right image in Figure 2 is a specific example of skill retrieval. In this example, the quadruped robot is assigned to complete a novel task, that is, the locomotion target on uneven ground is $K = (-1.2, 0, 0, 0)$. The entire skill retrieval process is divided into two parts. First, SG's most similar environment (indoor environment) is calculated by the Algorithm 1. Then, according to the similarity between the skills, we can find the most similar skill in SG, that is, the locomotion target is $K = (-1, 0, 0, 0)$.

Skill Reuse: We analyze the rapid learning and adaptation process of the robot from the perspective of return score and energy consumption. The Return and COT curves during the rapid learning process of the robot are shown in Figure 4 and Appendix Figure 9. We can find that, compared with the baseline algorithm SAC, our method has higher return scores and lower energy consumption in the stage of skill fine-tuning. Therefore, our method can make the robot learn new skills more stably from the original skills in the SG. The video in the supplementary material can further illustrate the effectiveness of our method. In contrast, the performance of the SAC algorithm fluctuates greatly, and it struggles to obtain a more stable skill within only 5000 steps.

The overall performance of the original and the fine-tuned skills in SG are shown in Figure 5. It can be seen that, when the new tasks are not too different from the skills in SG (such as tasks $11, 21$ and 33), the original skills have good generalization ability. It is similar to the return score of the fine-tuned skills. When the new tasks are quite different (such as tasks $31, 32$ and 34), the fine-tuned skill performance is greatly improved.

Furthermore, new skills learned by the robot will be added to the robotic SG, allowing the robot to continuously cope with the changing environment, as shown in Figure 6. Details are shown in Appendix Figure 13.

Figure 6: New skills (red nodes) have been added to robotic SG.

Table 2: Experimental results of our most relevant works. We list approximate numbers reported by the tasks most similar to ours. Specifically, we list the amount of real-world data used for training, and the associated wall-clock time (in minutes). Moreover, whether to utilize a simulation environment for training and whether to use an external network connection for real machine testing are considered. We also focus on comparing the skills learned in these studies.

Algorithms	Samples	Time	Simulation	External Connection	Learned Skills
Fine-Tuning	22.5×10^3	60	Yes	Yes	Recovery, forward and backward walking
Dreamer	72×10^3	60	No	Yes	Recovery, forward and backward walking
Efficient RL	20×10^3	20	No	Yes	Forward walking
Ours	$\mathbf{5 \times 10^3}$	$\mathbf{5 \sim 10}$	No	**No**	$\mathbf{844}$ skills covering a variety of desired locomotion targets and environments

Comparison with previous works: To further examine the significance of the robotic SG for DRL-based quadrupedal locomotion research, we compare it with some of the most relevant works, as shown in Table 2. These works all utilize the *Unitree A1* as the verification platform of the algorithm. Five aspects are being investigated, they are the sample number, the time used in real-world training, whether the training requires simulation, whether the execution of the policy requires an external network connection, and the specific skills learned by the policy.

It can be found that the sample number and time required by our proposed method are the least. Our method only needs about $5,000$ samples in the fine-tuning phase and can achieve stable performance on new tasks after training for 5 to 10 minutes. It means that robots can learn and adapt more quickly when faced with new tasks. We also do not need the simulation, thus bypassing the notorious reality gap problem (Koos et al., 2010). Meanwhile, the behavior of the robot without an external connection is more flexible, and we use *Wi-Fi* to communicate in real time. Whereas other work considers only a few robotic skills, we consider large-scale skills to achieve faster learning efficiency and better adaptability of robots. The SG greatly improves the scalability of the skills, laying the foundation for subsequent more challenging real-world robotics tasks.

6 Conclusion and Future Work

In this paper, we construct a novel robotic SG based on real-world quadrupedal robot skills to enable rapid learning and environmental adaptation. Different from common KG, SG mainly focuses on the dynamic behavior information of quadruped robots. To construct the SG, we designed the environment and tasks, and collected extensive empirical data based on real-world quadruped locomotion tasks. We then leveraged offline RL algorithm to obtain a representation of the robot's behavior, namely the policy and value network. Moreover, We defined entities, attributes, and build relations among them. The constructed SG supports functions such as visualization, skill retrieval, and skill reuse. Experiments on real-world tasks demonstrate the effectiveness of the SG for rapid learning of the robot's novel skills. In the future, we will continue to expand and improve SG, as shown in Appendix Figure 14. More robots with different mechanical structures, dynamic unstructured environments, and diverse skills will be considered. Although the robotic SG proposed is preliminary, it will be of great significance to the development of the DRL community (meta-RL, multi-task RL, offline RL, etc.), robotic learning, and other research fields.

References

Prithviraj Ammanabrolu and Mark O. Riedl. Learning knowledge graph-based world models of textual environments. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December*

6-14, 2021, virtual, pp. 3720–3731, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/1e747ddbea997a1b933aaf58a7953c3c-Abstract.html.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Christopher Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning. *CoRR*, abs/1912.06680, 2019. URL http://arxiv.org/abs/1912.06680.

Steven Bohez, Saran Tunyasuvunakool, Philemon Brakel, Fereshteh Sadeghi, Leonard Hasenclever, Yuval Tassa, Emilio Parisotto, Jan Humplik, Tuomas Haarnoja, Roland Hafner, et al. Imitate and repurpose: Learning reusable robot movement skills from human and animal behaviors. *arXiv preprint arXiv:2203.17138*, 2022.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Vinay K. Chaudhri, Chaitanya K. Baru, Naren Chittar, Xin Luna Dong, Michael R. Genesereth, James A. Hendler, Aditya Kalyanpur, Douglas B. Lenat, Juan Sequeda, Denny Vrandecic, and Kuansan Wang. Knowledge graphs: Introduction, history and, perspectives. *AI Mag.*, 43(1):17–29, 2022. doi: 10.1609/aimag.v43i1.19119. URL https://doi.org/10.1609/aimag.v43i1.19119.

Xinyue Chen, Che Wang, Zijian Zhou, and Keith W. Ross. Randomized ensembled double q-learning: Learning fast without a model. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=AY8zfZm0tDd.

Steve Collins, Andy Ruina, Russ Tedrake, and Martijn Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307(5712):1082–1085, 2005.

Carlo D'Eramo, Davide Tateo, Andrea Bonarini, Marcello Restelli, and Jan Peters. Sharing knowledge in multi-task deep reinforcement learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rkgpv2VFvr.

Ron Dorfman and Aviv Tamar. Offline meta reinforcement learning. *CoRR*, abs/2008.02598, 2020. URL https://arxiv.org/abs/2008.02598.

Ron Dorfman, Idan Shenfeld, and Aviv Tamar. Offline meta learning of exploration. *arXiv preprint arXiv:2008.02598*, 2020.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex X. Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *CoRR*, abs/1812.00568, 2018. URL http://arxiv.org/abs/1812.00568.

Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandeăić. Introducing wikidata to the linked data web. In *International Semantic Web Conference*, 2014.

Siddhant Gangapurwala, Mathieu Geisert, Romeo Orsolino, Maurice F. Fallon, and Ioannis Havoutis. Real-time trajectory adaptation for quadrupedal locomotion using deep reinforcement learning. In *IEEE International Conference on Robotics and Automation, ICRA 2021, Xi'an, China, May 30 - June 5, 2021*, pp. 5973–5979. IEEE, 2021. doi: 10.1109/ICRA48506.2021.9561639. URL https://doi.org/10.1109/ICRA48506.2021.9561639.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1856–1865. PMLR, 2018. URL http://proceedings.mlr.press/v80/haarnoja18b.html.

Danijar Hafner, Timothy P. Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 2019. URL http://proceedings.mlr.press/v97/hafner19a.html.

Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Takuya Hiraoka, Takahisa Imagawa, Taisei Hashimoto, Takashi Onishi, and Yoshimasa Tsuruoka. Dropout q-functions for doubly efficient reinforcement learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=xCVJMsPv3RT.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, Roberto Navigli, Axel-Cyrille Ngonga Ngomo, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge Graphs. *arXiv e-prints*, art. arXiv:2003.02320, March 2020.

Sunghoon Hong, Deunsol Yoon, and Kee-Eung Kim. Structure-aware transformer policy for inhomogeneous multi-task reinforcement learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=fy_XRVHqly.

Niklas Höpner, Ilaria Tiddi, and Herke van Hoof. Leveraging class abstraction for commonsense reinforcement learning via residual policy gradient methods. In Luc De Raedt (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pp. 3050–3056. ijcai.org, 2022. doi: 10.24963/ijcai.2022/423. URL https://doi.org/10.24963/ijcai.2022/423.

Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26), 2019.

Atil Iscen, Ken Caluwaerts, Jie Tan, Tingnan Zhang, Erwin Coumans, Vikas Sindhwani, and Vincent Vanhoucke. Policies modulating trajectory generators. In *2nd Annual Conference on Robot Learning, CoRL 2018, Zürich, Switzerland, 29-31 October 2018, Proceedings*, volume 87 of *Proceedings of Machine Learning Research*, pp. 916–926. PMLR, 2018. URL http://proceedings.mlr.press/v87/iscen18a.html.

Deepali Jain, Atil Iscen, and Ken Caluwaerts. Hierarchical reinforcement learning for quadruped locomotion. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2019, Macau, SAR, China, November 3-8, 2019*, pp. 7551–7557. IEEE, 2019. doi: 10.1109/IROS40897.2019.8967913. URL https://doi.org/10.1109/IROS40897.2019.8967913.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Networks Learn. Syst.*, 33(2):494–514, 2022. doi: 10.1109/TNNLS.2021.3070843. URL https://doi.org/10.1109/TNNLS.2021.3070843.

Minqi Jiang, Jelena Luketina, Nantas Nardelli, Pasquale Minervini, Philip H. S. Torr, Shimon Whiteson, and Tim Rocktäschel. Wordcraft: An environment for benchmarking commonsense agents. *CoRR*, abs/2007.09185, 2020. URL https://arxiv.org/abs/2007.09185.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. Scalable deep reinforcement learning for vision-based robotic manipulation. In *2nd Annual Conference on*

Robot Learning, CoRL 2018, Zürich, Switzerland, 29-31 October 2018, Proceedings, volume 87 of *Proceedings of Machine Learning Research*, pp. 651–673. PMLR, 2018. URL http://proceedings.mlr.press/v87/kalashnikov18a.html.

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of generalisation in deep reinforcement learning. *CoRR*, abs/2111.09794, 2021. URL https://arxiv.org/abs/2111.09794.

Sylvain Koos, Jean-Baptiste Mouret, and Stéphane Doncieux. Crossing the Reality Gap in Evolutionary Robotics by Promoting Transferable Controllers. In *Conference on Genetic and Evolutionary Computation*, pp. 119–126, United States, July 2010. ACM, publisher. URL https://hal.archives-ouvertes.fr/hal-00633927.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0d2b2061826a5df3221116a5085a6052-Abstract.html.

Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47), 2020. doi: 10.1126/scirobotics.abc5986. URL https://robotics.sciencemag.org/content/5/47/eabc5986.

Youngwoon Lee, Joseph J. Lim, Anima Anandkumar, and Yuke Zhu. Adversarial skill chaining for long-horizon robot manipulation via terminal state regularization. In Aleksandra Faust, David Hsu, and Gerhard Neumann (eds.), *Conference on Robot Learning, 8-11 November 2021, London, UK*, volume 164 of *Proceedings of Machine Learning Research*, pp. 406–416. PMLR, 2021. URL https://proceedings.mlr.press/v164/lee22a.html.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Social Work*, 6: 167–195, 2015.

Lanqing Li, Rui Yang, and Dijun Luo. FOCAL: efficient fully-offline meta-reinforcement learning via distance metric learning and behavior regularization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=8cpHIfgY4Dj.

Linfeng Li, Peng Wang, Jun Yan, Yao Wang, Simin Li, Jinpeng Jiang, Zhe Sun, Buzhou Tang, Tsung-Hui Chang, Shenghui Wang, et al. Real-world data medical knowledge graph: construction and applications. *Artificial intelligence in medicine*, 103:101817, 2020.

Sen Lin, Jialin Wan, Tengyu Xu, Yingbin Liang, and Junshan Zhang. Model-based offline meta-reinforcement learning with regularization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=EBn0uInJZWh.

F. Mahdisoltani, J. Biega, and F. M. Suchanek. Yago3: A knowledge base from multilingual wikipedias. In *Conference on Innovative Data Systems Research*, 2014.

Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022. doi: 10.1126/scirobotics.abk2822. URL https://www.science.org/doi/abs/10.1126/scirobotics.abk2822.

Eric Mitchell, Rafael Rafailov, Xue Bin Peng, Sergey Levine, and Chelsea Finn. Offline meta-reinforcement learning with advantage weighting. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7780–7791. PMLR, 2021. URL http://proceedings.mlr.press/v139/mitchell21a.html.

Keerthiram Murugesan, Mattia Atzeni, Pavan Kapanipathi, Pushkar Shukla, Sadhana Kumaravel, Gerald Tesauro, Kartik Talamadupula, Mrinmaya Sachan, and Murray Campbell. Text-based RL agents with commonsense knowledge: New challenges, environments and baselines. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 9018–9027. AAAI Press, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17090.

Ofir Nachum, Haoran Tang, Xingyu Lu, Shixiang Gu, Honglak Lee, and Sergey Levine. Why does hierarchy (sometimes) work so well in reinforcement learning? *CoRR*, abs/1909.10618, 2019. URL http://arxiv.org/abs/1909.10618.

Taewook Nam, Shao-Hua Sun, Karl Pertsch, Sung Ju Hwang, and Joseph J. Lim. Skill-based meta-reinforcement learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=jeLW-Fh9bV.

Peter Pastor, Heiko Hoffmann, Tamim Asfour, and Stefan Schaal. Learning and generalization of motor skills by learning from demonstration. In *2009 IEEE International Conference on Robotics and Automation, ICRA 2009, Kobe, Japan, May 12-17, 2009*, pp. 763–768. IEEE, 2009. doi: 10.1109/ROBOT.2009.5152385. URL https://doi.org/10.1109/ROBOT.2009.5152385.

Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Edward Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. In *Robotics: Science and Systems*, 07 2020. doi: 10.15607/RSS.2020.XVI.064.

Karl Pertsch, Youngwoon Lee, and Joseph J. Lim. Accelerating reinforcement learning with learned skill priors. In Jens Kober, Fabio Ramos, and Claire J. Tomlin (eds.), *4th Conference on Robot Learning, CoRL 2020, 16-18 November 2020, Virtual Event / Cambridge, MA, USA*, volume 155 of *Proceedings of Machine Learning Research*, pp. 188–204. PMLR, 2020. URL https://proceedings.mlr.press/v155/pertsch21a.html.

Karl Pertsch, Youngwoon Lee, Yue Wu, and Joseph J. Lim. Demonstration-guided reinforcement learning with learned skills. In Aleksandra Faust, David Hsu, and Gerhard Neumann (eds.), *Conference on Robot Learning, 8-11 November 2021, London, UK*, volume 164 of *Proceedings of Machine Learning Research*, pp. 729–739. PMLR, 2021. URL https://proceedings.mlr.press/v164/pertsch22a.html.

Vitchyr H. Pong, Ashvin V. Nair, Laura M. Smith, Catherine Huang, and Sergey Levine. Offline meta-reinforcement learning with online self-supervision. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17811–17829. PMLR, 2022. URL https://proceedings.mlr.press/v162/pong22a.html.

Maurice Rahme, Ian Abraham, Matthew L. Elwin, and Todd D. Murphey. Dynamics and domain randomized gait modulation with bezier curves for sim-to-real legged locomotion. *CoRR*, abs/2010.12070, 2020. URL https://arxiv.org/abs/2010.12070.

Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5331–5340. PMLR, 2019. URL http://proceedings.mlr.press/v97/rakelly19a.html.

Dushyant Rao, Fereshteh Sadeghi, Leonard Hasenclever, Markus Wulfmeier, Martina Zambelli, Giulia Vezzani, Dhruva Tirumala, Yusuf Aytar, Josh Merel, Nicolas Heess, and Raia Hadsell. Learning transferable motor skills with hierarchical latent mixture policies. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=qTHBE7E9iej.

Sasha Salter, Kristian Hartikainen, Walter Goodwin, and Ingmar Posner. Priors, hierarchy, and information asymmetry for skill transfer in reinforcement learning. *CoRR*, abs/2201.08115, 2022. URL https://arxiv.org/abs/2201.08115.

Tanmay Shankar and Abhinav Gupta. Learning robot skills with temporal variational inference. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8624–8633. PMLR, 2020. URL http://proceedings.mlr.press/v119/shankar20b.html.

Tanmay Shankar, Yixin Lin, Aravind Rajeswaran, Vikash Kumar, Stuart Anderson, and Jean Oh. Translating robot skills: Learning unsupervised skill correspondences across robots. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 19626–19644. PMLR, 2022. URL https://proceedings.mlr.press/v162/shankar22a.html.

Haojie Shi, Bo Zhou, Hongsheng Zeng, Fan Wang, Yueqiang Dong, Jiangyong Li, Kang Wang, Hao Tian, and Max Q.-H. Meng. Reinforcement learning with evolutionary trajectory generator: A general approach for quadrupedal locomotion. *IEEE Robotics Autom. Lett.*, 7(2):3085–3092, 2022a. doi: 10.1109/LRA.2022.3145495. URL https://doi.org/10.1109/LRA.2022.3145495.

Lucy Xiaoyang Shi, Joseph J. Lim, and Youngwoon Lee. Skill-based model-based reinforcement learning. *CoRR*, abs/2207.07560, 2022b. doi: 10.48550/arXiv.2207.07560. URL https://doi.org/10.48550/arXiv.2207.07560.

Edward Shortliffe. *Computer-based medical consultations: MYCIN*, volume 2. Elsevier, 2012.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nat.*, 529(7587):484–489, 2016. doi: 10.1038/nature16961. URL https://doi.org/10.1038/nature16961.

Abhik Singla, Shounak Bhattacharya, Dhaivat Dholakiya, Shalabh Bhatnagar, Ashitava Ghosal, Bharadwaj Amrutur, and Shishir Kolathaya. Realizing learned quadruped locomotion behaviors through kinematic motion primitives. In *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*, pp. 7434–7440. IEEE, 2019. doi: 10.1109/ICRA.2019.8794179. URL https://doi.org/10.1109/ICRA.2019.8794179.

Samarth Sinha, Ajay Mandlekar, and Animesh Garg. S4RL: surprisingly simple self-supervision for offline reinforcement learning in robotics. In Aleksandra Faust, David Hsu, and Gerhard Neumann (eds.), *Conference on Robot Learning, 8-11 November 2021, London, UK*, volume 164 of *Proceedings of Machine Learning Research*, pp. 907–917. PMLR, 2021. URL https://proceedings.mlr.press/v164/sinha22a.html.

Laura M. Smith, J. Chase Kew, Xue Bin Peng, Sehoon Ha, Jie Tan, and Sergey Levine. Legged robots that keep on learning: Fine-tuning locomotion policies in the real world. In *2022 International Conference on Robotics and Automation, ICRA 2022, Philadelphia, PA, USA, May 23-27, 2022*, pp. 1593–1599. IEEE, 2022a. doi: 10.1109/ICRA46639.2022.9812166. URL https://doi.org/10.1109/ICRA46639.2022.9812166.

Laura M. Smith, Ilya Kostrikov, and Sergey Levine. A walk in the park: Learning to walk in 20 minutes with model-free reinforcement learning. *CoRR*, abs/2208.07860, 2022b. doi: 10.48550/arXiv.2208.07860. URL https://doi.org/10.48550/arXiv.2208.07860.

Shagun Sodhani, Amy Zhang, and Joelle Pineau. Multi-task reinforcement learning with context-based representations. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9767–9779. PMLR, 2021. URL http://proceedings.mlr.press/v139/sodhani21a.html.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.*, 112(1-2):181–211, 1999. doi: 10.1016/S0004-3702(99)00052-1. URL https://doi.org/10.1016/S0004-3702(99)00052-1.

Michita Imai Takuma Seno. d3rlpy: An offline deep reinforcement library. In *NeurIPS 2021 Offline Reinforcement Learning Workshop*, December 2021.

Mathias Thor, Tomas Kulvicius, and Poramate Manoonpong. Generic neural locomotion control framework for legged robots. *IEEE Transactions on Neural Networks and Learning Systems*, 32 (9):4013–4025, 2021. doi: 10.1109/TNNLS.2020.3016523.

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander Sasha Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom Le Paine, Çaglar Gülçehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy P. Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in starcraft II using multi-agent reinforcement learning. *Nat.*, 575(7782):350–354, 2019. doi: 10.1038/s41586-019-1724-z. URL https://doi.org/10.1038/s41586-019-1724-z.

Eric Vollenweider, Marko Bjelonic, Victor Klemm, Nikita Rudin, Joonho Lee, and Marco Hutter. Advanced skills through multiple adversarial motion priors in reinforcement learning. *arXiv preprint arXiv:2203.14912*, 2022.

Qi Wang and Herke van Hoof. Model-based meta reinforcement learning using graph structured surrogate models and amortized policy search. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23055–23077. PMLR, 2022. URL https://proceedings.mlr.press/v162/wang22z.html.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, and Pieter Abbeel. Daydreamer: World models for physical robot learning. *CoRR*, abs/2206.14176, 2022. doi: 10.48550/arXiv.2206.14176. URL https://doi.org/10.48550/arXiv.2206.14176.

Mengdi Xu, Yikang Shen, Shun Zhang, Yuchen Lu, Ding Zhao, Joshua B. Tenenbaum, and Chuang Gan. Prompting decision transformer for few-shot policy generalization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 24631–24645. PMLR, 2022. URL https://proceedings.mlr.press/v162/xu22g.html.

Chuanyu Yang, Kai Yuan, Qiuguo Zhu, Wanming Yu, and Zhibin Li. Multi-expert learning of adaptive legged locomotion. *Science Robotics*, 5(49), 2020a. doi: 10.1126/scirobotics.abb2174. URL https://robotics.sciencemag.org/content/5/49/eabb2174.

Ruihan Yang, Huazhe Xu, Yi Wu, and Xiaolong Wang. Multi-task reinforcement learning with soft modularization. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020b. URL https://proceedings.neurips.cc/paper/2020/hash/32cfdce9631d8c7906e8e9d6e68b514b-Abstract.html.

Yuxiang Yang, Tingnan Zhang, Erwin Coumans, Jie Tan, and Byron Boots. Fast and efficient locomotion via learned gait transitions. In Aleksandra Faust, David Hsu, and Gerhard Neumann (eds.), *Conference on Robot Learning, 8-11 November 2021, London, UK*, volume 164 of *Proceedings of Machine Learning Research*, pp. 773–783. PMLR, 2021. URL https://proceedings.mlr.press/v164/yang22d.html.

Qingfeng Yao, Jilong Wang, Donglin Wang, Shuyu Yang, Hongyin Zhang, Yinuo Wang, and Zhengqing Wu. Hierarchical terrain-aware control for quadrupedal locomotion by combining deep reinforcement learning and optimal control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - Oct. 1, 2021*, pp. 4546–4551. IEEE, 2021. doi: 10.1109/IROS51168.2021.9636738. URL `https://doi.org/10.1109/IROS51168.2021.9636738`.

Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Sergey Levine, and Chelsea Finn. Conservative data sharing for multi-task offline reinforcement learning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 11501–11516, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/5fd2c06f558321eff612bbbe455f6fbd-Abstract.html`.

Haoqi Yuan and Zongqing Lu. Robust task representations for offline meta-reinforcement learning via contrastive learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 25747–25759. PMLR, 2022. URL `https://proceedings.mlr.press/v162/yuan22a.html`.

Chicheng Zhang and Zhi Wang. Provably efficient multi-task reinforcement learning with model transfer. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 19771–19783, 2021. URL `https://proceedings.neurips.cc/paper/2021/hash/a440a3d316c5614c7a9310e902f4a43e-Abstract.html`.

Hongyin Zhang, Jilong Wang, Zhengqing Wu, Yinuo Wang, and Donglin Wang. Terrain-aware risk-assessment-network-aided deep reinforcement learning for quadrupedal locomotion in tough terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - Oct. 1, 2021*, pp. 4538–4545. IEEE, 2021. doi: 10.1109/IROS51168.2021.9636519. URL `https://doi.org/10.1109/IROS51168.2021.9636519`.

Feng Zhao, Ziqi Zhang, and Donglin Wang. Ksg: Knowledge and skill graph. 2022.

A APPENDIX

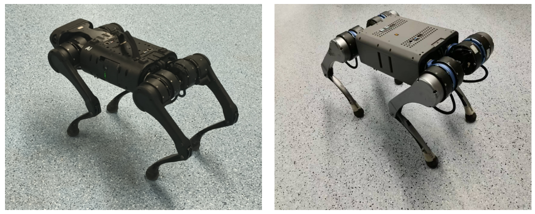


Figure 7: Two robots were used in the initial release of SG: the *Unitree A1* robot (Left) and our own robot (Right).

Table 3: The dynamic and kinematic parameters of robots. These values will be used as attributes of the robot entities. Robot 1 is *Unitree A1*, and Robot 2 is made by ourselves. Parameters $I_{xx}, I_{xy}, I_{xz}, I_{yy}, I_{yz}$ and I_{zz} are the approximated inertia for the single rigid body dynamic model. The International System of Units is used.

Dynamic and Kenimatic Paramteres		
Parameters	Robot 1	Robot 2
Trunk length	0.361 m	0.28 m
Trunk width	0.094 m	0.2 m
Trunk weight	6 kg	4.953 kg
Hip link length	0.0838 m	0.04 m
Hip link weight	0.696 kg	0.54 kg
Thigh link length	0.2 m	0.2 m
Thigh link weight	1.013 kg	0.886 kg
Calf link length	0.2 m	0.2 m
Calf link weight	0.166 kg	0.119 kg
Total Mass	13.74 kg	11.149 kg
I_{xx}	0.016 kgm^2	0.010 kgm^2
I_{xy}	-3.66×10^{-5} kgm^2	1.608×10^{-6} kgm^2
I_{xz}	-6.11×10^{-5} kgm^2	6.104×10^{-6} kgm^2
I_{yy}	0.038 kgm^2	0.011 kgm^2
I_{yz}	-2.75×10^{-5} kgm^2	2.358×10^{-6} kgm^2
I_{zz}	0.046 kgm^2	0.017 kgm^2

Table 4: Environmental parameters. These values will be used as attributes of the environment entity.

Environmental Parameters		
Environments	Friction Coefficient	Slope Angle (rad)
Indoor Marble Flat Floor	$0.25 \sim 0.5$	0
Outdoor Marble Flat Floor	$0.4 \sim 0.6$	0
Marble Slope Floor	$0.4 \sim 0.6$	0.174
Asphalt Road	0.72	0
Grassland	0.35	0

Table 5: In designing locomotion targets, we select 4 different controllable variables in robot states to be the desired states in robot tasks. X, Y, A, and P are the index of $v_x, v_y, d\psi$ and P, respectively.

X	$v_x(m/s)$	Y	$v_y(m/s)$	A	$d\psi(rad/s)$	P	$\theta(rad)$
0	0	0	0	0	0	0	0
1	-1	1	-0.8	1	-2	1	-0.4
2	-0.75	2	-0.6	2	-1.5	2	-0.2
3	-0.5	3	-0.4	3	-1	3	0.2
4	-0.25	4	-0.2	4	-0.5	4	0.4
5	0.25	5	0.2	5	0.5		
6	0.5	6	0.4	6	1		
7	0.75	7	0.6	7	1.5		
8	1	8	0.8	8	2		

Table 6: The variables are paired together as the reference state of the task while the rest of the variables are set to be their default value.

	$v_x(m/s)$	$v_y(m/s)$	$d\psi(rad/s)$	$\theta(rad)$
v_x	$X000$	$XY00$	$X0A0$	$X00P$
v_y	-	$0Y00$	$0YA0$	$X00P$
$d\psi$	-	-	$00A0$	$00AP$

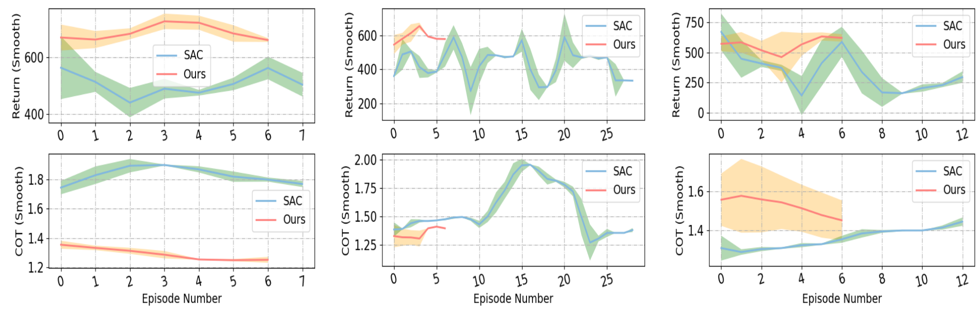
Figure 8: Three new environments are considered in the skill reuse phase of robotic SG, including EVA foam floor mats, sponge mat, and wooden boards respectively.

Table 7: Four novel locomotion targets are considered in the skill reuse phase of robotic SG.

	$v_x(m/s)$	$v_y(m/s)$	$d\psi(rad/s)$	$\theta(rad)$
1	0.9	0	0	0
2	1.2	0	0.6	0
3	-1.2	0	0	0
4	1.2	0.48	0	0

Table 8: Hyperparameters used for CQL and SAC algorithms. Other unspecified hyperparameters are the same as Takuma Seno (2021).

Algorithms	Hyperparameters
CQL	actor_encoder = MLP (hidden_units=[256, 256], activation='tanh') critic_encoder = MLP (hidden_units=[256, 256], activation='tanh') conservative_weight=0.1
SAC	actor_encoder = MLP (hidden_units=[256, 256], activation='tanh') critic_encoder = MLP (hidden_units=[256, 256], activation='tanh') batch_size=512 n_steps=4



(a) Task 11 (b) Task 21 (c) Task 31

Figure 9: Return (upper) and COT (bottom) of the rapid learning and adaptation process of quadruped robots on new real-world tasks 11, 21 and 31. Task MN represents the new task number, where M represents the M^{th} environment in Figure 8 , and N represents the N^{th} locomotion target in Table 7. The x-axis represents the number of episodes. The shaded area represents one standard deviation. Each experiment was repeated three times.

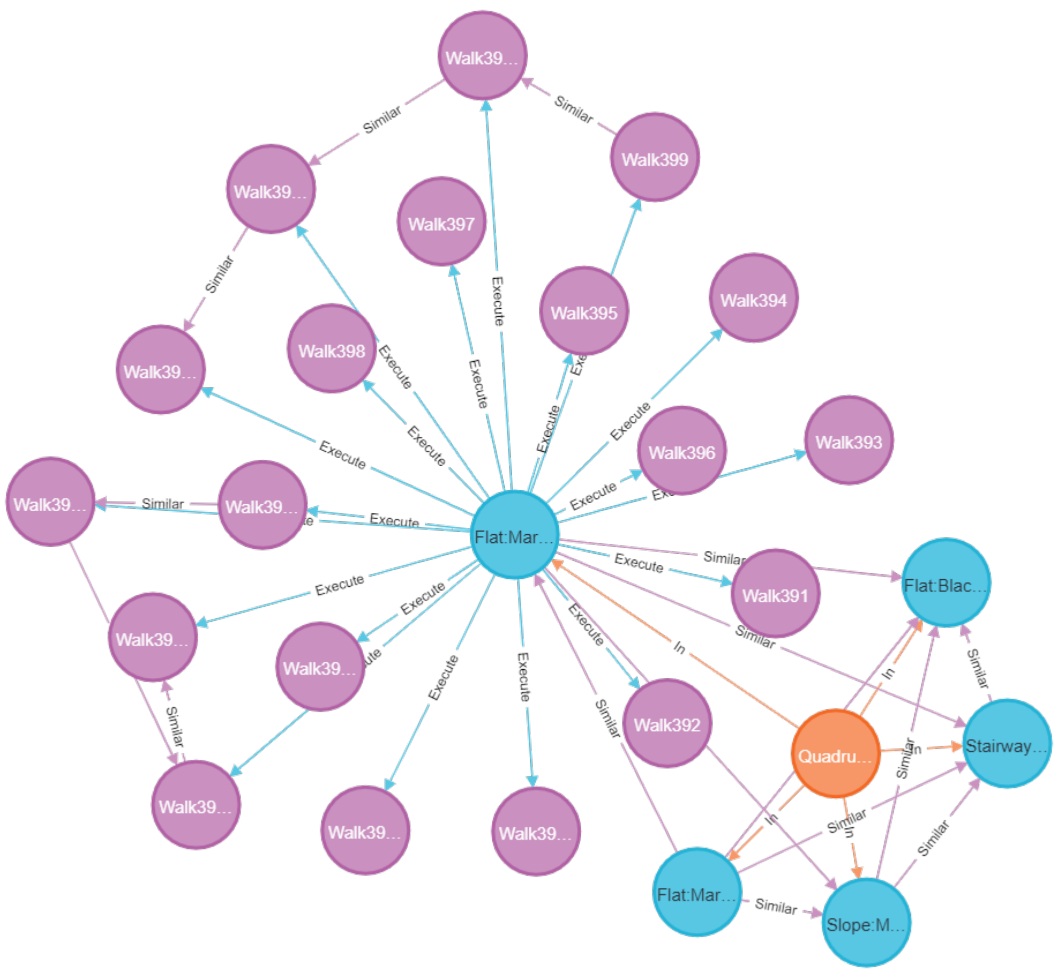


Figure 10: Visual details of relations between entities, and relations between environments.

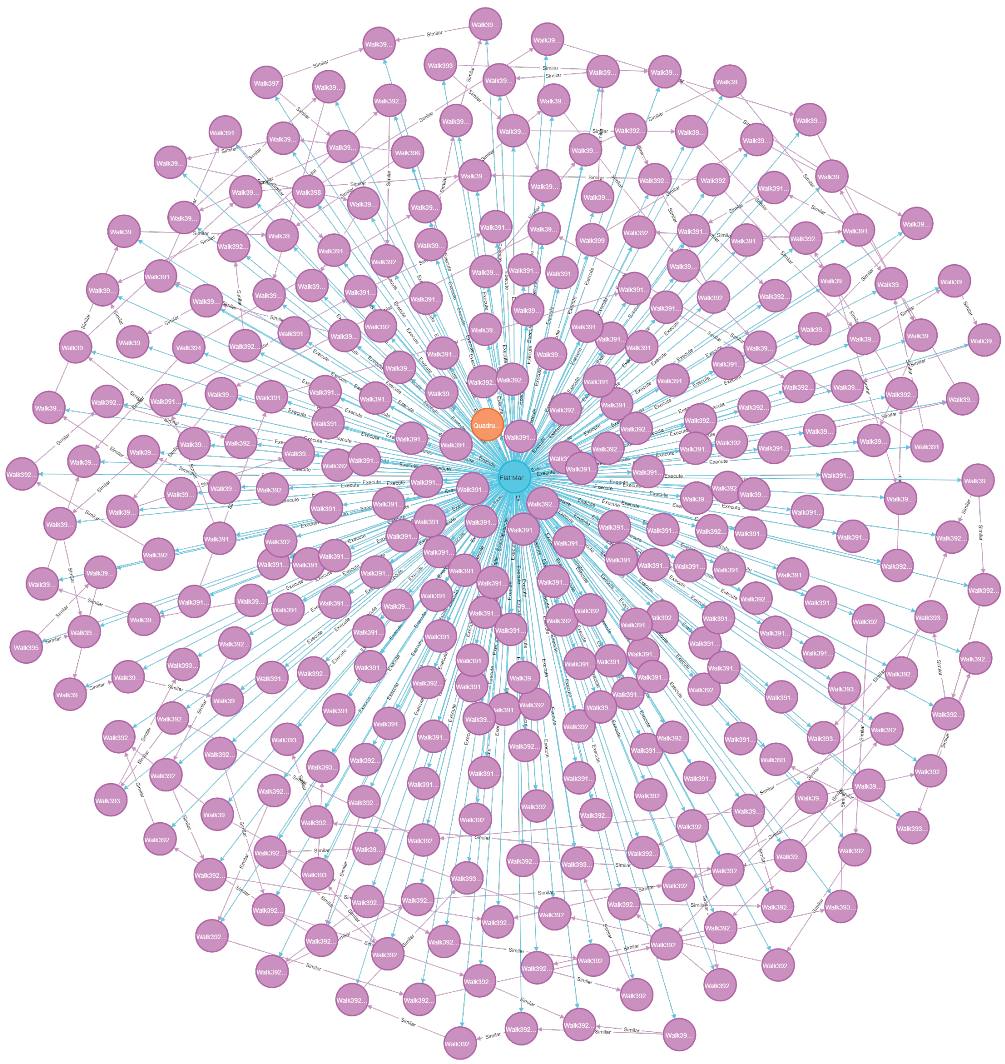


Figure 11: Visualization of all the skills of a robot in an environment.

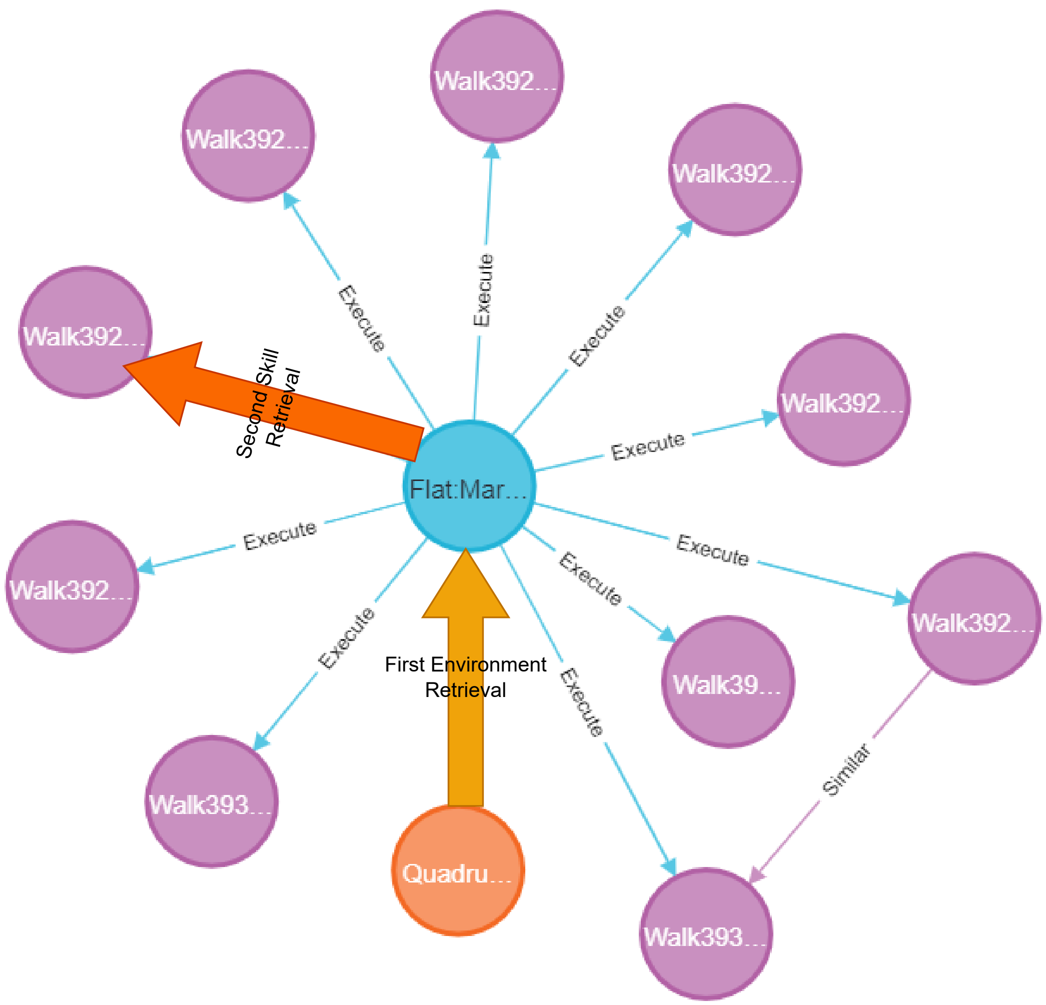


Figure 12: Visual details of the skill retrieval process.

Figure 13: New skills (red nodes) have been added to robotic SG in detail.

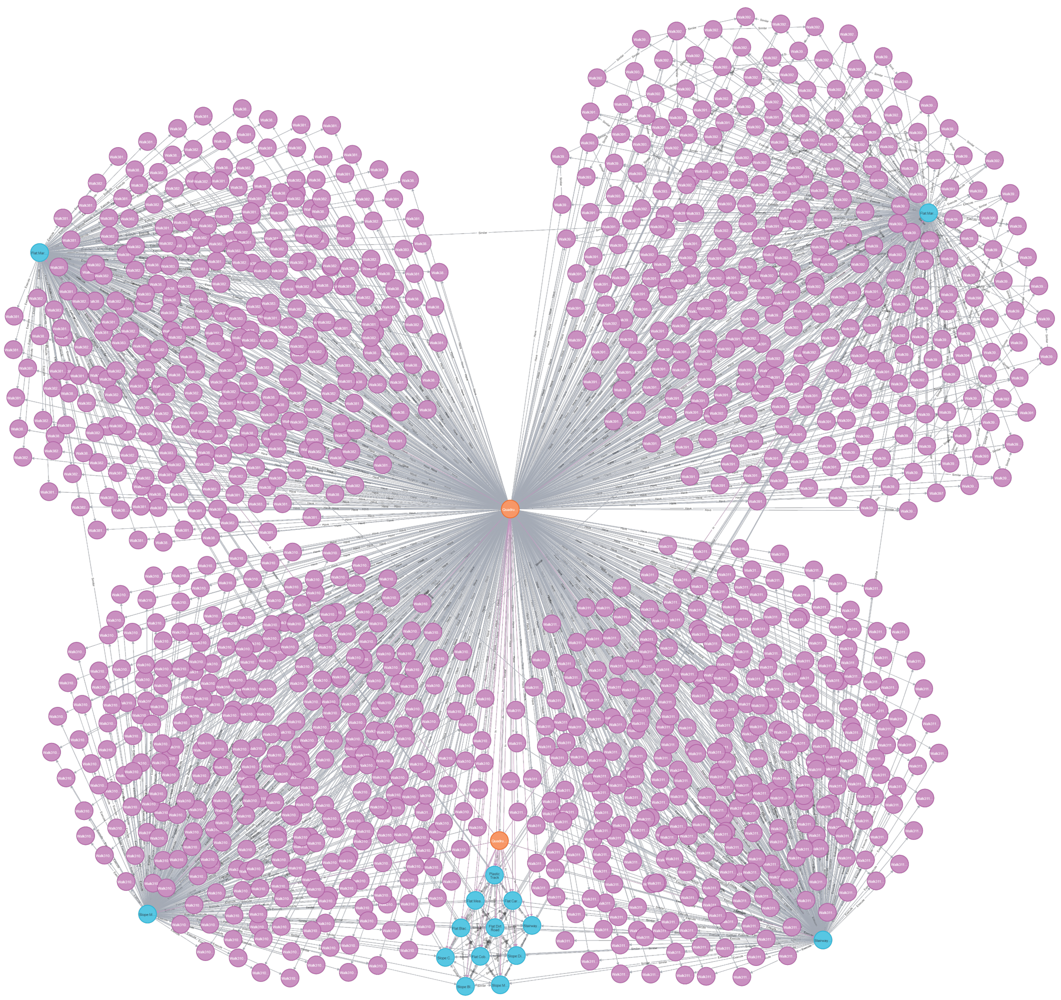


Figure 14: A visualization of the SG we plan to complete in the near future. It is expected to contain dozens of common environments and thousands of robotic skills.