MoE-Loco: Mixture of Experts for Multitask Locomotion

Runhan Huang*[1,2], Shaoting Zhu*[1,2], Yilun Du[3], Hang Zhao†[1,2]

MoE-Loco.github.io/

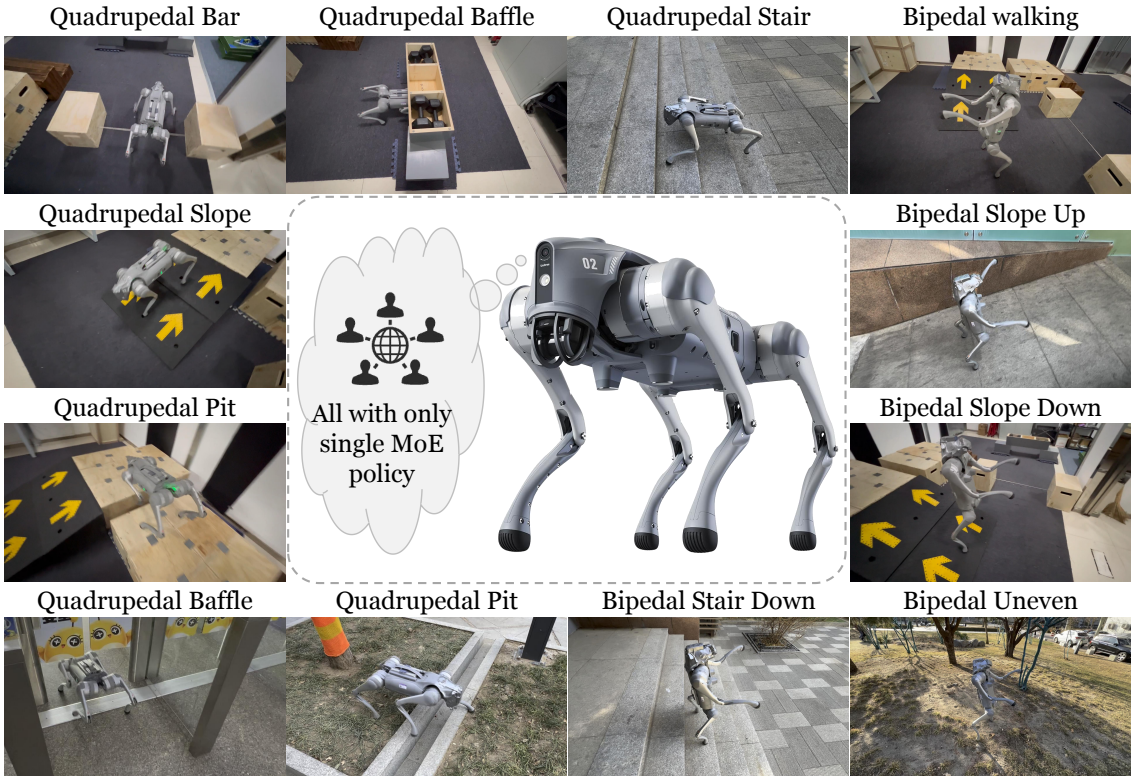

Fig. 1: **We introduce MoE-Loco.** With a single MoE policy, quadruped robot can traverse a variety of challenging terrains and perform different locomotion modes, including bipedal and quadrupedal gaits.

***Abstract*— We present MoE-Loco, a Mixture of Experts (MoE) framework for multitask locomotion for legged robots. Our method enables a single policy to handle diverse terrains, including bars, pits, stairs, slopes, and baffles, while supporting quadrupedal and bipedal gaits. Using MoE, we mitigate the gradient conflicts that typically arise in multitask reinforcement learning, improving both training efficiency and performance. Our experiments demonstrate that different experts naturally specialize in distinct locomotion behaviors, which can be leveraged for task migration and skill composition. We further validate our approach in both simulation and real-world deployment, showcasing its robustness and adaptability.**

I. INTRODUCTION

Robots are often required to traverse diverse terrains and demonstrate various skills [1], [2]. Recent advancements in reinforcement learning (RL) algorithms and physics-based simulators have enabled RL-based approaches to become the dominant paradigm for training robot locomotion policies [3]–[7]. However, while single-task RL has demonstrated remarkable success, learning a unified policy that generalizes across multiple tasks, terrains, and locomotion modes remains a significant challenge.

Recent research has explored training locomotion policies with diverse skills by having diverse terrains in simulation for parallel training [8]–[10]. However, multitask RL with a simple neural network architecture often suffers from gradient conflicts [11], [12], which leads to inferior model performance. What is worse, training a policy across multiple terrains with different gaits poses further challenges, leading to model divergence.

In this work, we enable a quadruped robot to traverse various terrains—including bars, pits, stairs, slopes, and baffles—while also supporting gait switching between bipedal and quadrupedal modes, using only one policy. We integrate the Mixture of Experts (MoE) framework [13]–[16] as a modular network structure for multitask locomotion reinforcement learning. We demonstrate that the MoE framework alleviates gradient conflicts by directing gradients to specialized experts, thus improving training efficiency and overall performance. Furthermore, we analyze the roles of

[1]IIIS, Tsinghua University, Beijing, China
[2]Shanghai Qi Zhi Institute, Shanghai, China
[3]Harvard University, MA, USA
* These authors contributed equally to this work.
† Corresponding at: hangzhao@mail.tsinghua.edu.cn

different experts within the MoE model and observe that they naturally specialize in distinct behaviors. Leveraging this property, we can manually adjust the ratios between different experts to compose new skills, underscoring the adaptability and reusability of our modular approach for novel tasks.

In summary, our contributions are as follows.

- We train and deploy a single neural network policy that enables a quadruped robot to **cross challenging terrains** and perform fundamentally different locomotion modes, including **bipedal and quadrupedal gaits**.
- We integrate the **MoE architecture** into locomotion policy training to **mitigate gradient conflicts**, improve training efficiency, and overall model performance.
- We conduct qualitative and quantitative analysis of MoE, uncovering expert specialization patterns. Using these insights, we explore the potential of MoE for **task migration** and **skill composition**.

II. RELATED WORKS

A. Reinforcement Learning for Robot Locomotion

Using reinforcement learning for robot locomotion control has become increasingly popular in recent years. It has demonstrated the ability to learn legged locomotion behaviors in both simulation [17], [18] and the real world [3], [19]. Not only can it traverse a variety of complex terrains [8], [20], [21], but it can also achieve high-speed running [22], [23]. Furthermore, reinforcement learning has enabled robots to perform extreme tasks and master skills such as bipedal walking [24], [25], opening doors [1], [4], navigating rocky terrains [26], and even executing high-speed parkour in challenging environments [7], [9], [10], [27]. However, most of these works focus on specific skills and a limited number of terrains, rarely considering multitask learning.

B. MultiTask Learning

Multitask learning (MTL) aims to train a unified network that can perform across different tasks [28]–[30]. MTL allows multiple tasks to benefit from shared knowledge [31], [32], but some works highlight the challenge of negative gradient conflicts during training [33]–[35]. Multitask Reinforcement Learning (MTRL) is one of the most popular areas of research in this domain. Many algorithms have been developed to improve the effectiveness of MTRL [36]–[38]. Moreover, MTRL is widely applied in the robotics field, though much of the focus has been on manipulation tasks [16], [35], [39], [40]. In the context of locomotion, works like ManyQuadrupeds [41] focus on learning a unified policy for different categories of quadruped robots. MELA [42] employs pretrained expert models to construct a locomotion policy, although it primarily concentrates on basic skill acquisition. Moreover, the pretraining process for specialized neural network policies requires substantial reward engineering efforts. MTAC [43] attempts to train a policy across different terrains using hierarchical RL, but their approach can only handle one gait with three relatively simple terrains and has not been deployed on a real robot.

C. Mixture of Experts (MoE)

The concept of Mixture of Experts (MoE), originally introduced in [13], [44], has received extensive attention in recent years [45], [46]. It has found widespread application in fields such as natural language processing [47], [48], computer vision [49], [50], and multi-modal learning [51], [52]. MoE has also been applied in reinforcement learning and robotics. DeepMind [14] has explored using MoE to scale reinforcement learning. MELA [39] proposed a Multi-Expert Learning Architecture to generate adaptive skills from a set of representative expert skills, but their focus is primarily on simple actions.

III. METHOD

A. Task Definition

In this paper, we focus on 9 challenging locomotion tasks, encompassing both quadrupedal and bipedal gaits. The quadrupedal gait tasks include bar crossing, pit crossing, baffle crawling, stair climbing, and slope walking. The bipedal gait tasks consist of standing up, plane walking, slope walking, and stair descending. Our terrains in the simulation environment are shown in Figure 2. The robot is controlled via velocity commands from a joystick, where a one-hot vector is used to indicate whether to walk in the quadrupedal or bipedal gait.

Fig. 2: **A snapshot of the terrain settings.** From left to right: bar, pit, baffle, slope, and stairs.

We define multitask locomotion as a Markov Decision Process (MDP), represented by the tuple $\langle S_\tau, A_\tau, T_\tau, R_\tau, \gamma_\tau \rangle$. In our training framework, the multitask nature is characterized by several key aspects. First, different locomotion terrains correspond to distinct subsets of the state space, denoted as S_τ, where $S_\tau \subseteq S$ represents the states relevant to a specific task. However, the full state space S remains unknown to the robot. For instance, the task of walking on slopes involves a state space that differs from those required for stair climbing or bar traversal.

Moreover, the reward function R varies across different gaits, reflecting task-specific objectives. Additionally, the termination conditions depend on the type of gait, leading to distinct transition dynamics T. The robot learns a policy $\pi(a|s)$ that selects actions based on both the terrain and gait, aiming to maximize the cumulative reward across tasks:

$$J(\pi) = \mathbb{E}\left[\sum_{\tau}\sum_{t=0}^{\infty}\gamma^t R_\tau(s_t, a_t, s_{t+1})\right] \tag{1}$$

The goal is to learn a single universal policy that generalizes across various tasks. We demonstrate our method in the condition of blind locomotion (only use proprioception as input). In fact, our approach can also be incorporated into visual RL locomotion settings, further enhancing their multitask locomotion capabilities.

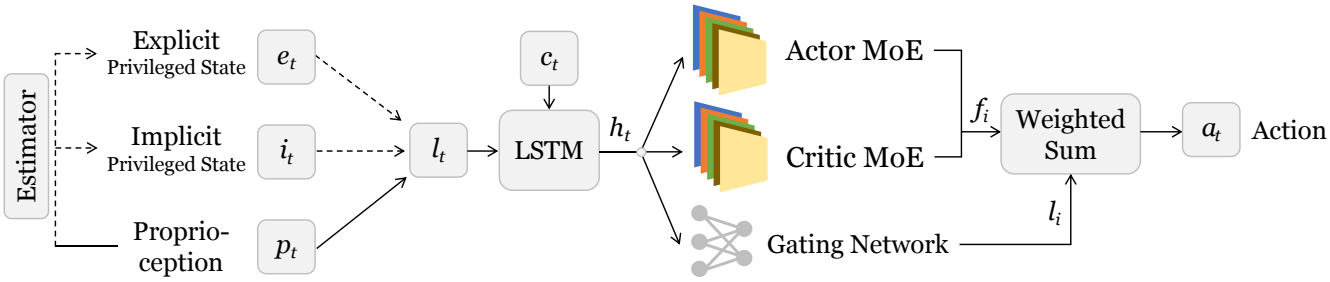

Fig. 3: **Overview of our MoELoco pipeline.** With the design of MoE architecture, our policy achieves robust multitask locomotion ability on various challenging terrains with multiple gaits.

B. MoE Based Multitask Locomotion Learning

In this section, we introduce our MoE based multitask locomotion learning. We incorporate a two-stage training framework following [20], and use PPO [53] as our reinforcement learning algorithm.

State Space: The entire process includes four types of observations: proprioception $\boldsymbol{p}_t$, explicit privileged state $\boldsymbol{e}_t$, implicit privileged state $\boldsymbol{i}_t$, and command $\boldsymbol{c}_t$. **1) Proprioception $\boldsymbol{p}_t$** includes projected gravity and base angular velocity from the IMU, joint positions, joint velocities, and the last action. **2) Explicit privileged state $\boldsymbol{e}_t$** contains the base linear velocity (IMU data is too noisy to use) and ground friction. **3) Implicit privileged state $\boldsymbol{i}_t$** includes contact force of different robot link, which must be encoded into a low-dimensional latent representation to mitigate the sim-to-real gap [20]. **4) Command $\boldsymbol{c}_t$** consists of a velocity command $V = (v_x, v_y, v_{\text{yaw}})$ and a one-hot vector g, where $g = 0$ represents a quadrupedal gait and $g = 1$ represents a bipedal gait in the context of multitask reinforcement learning.

Action Space: The action space $\boldsymbol{a}_t \in \mathbb{R}^{12}$ consists of the desired joint positions for all 12 joints.

Reward Design: Under the setting of the multitask learning, the robot receives different rewards based on the gait command g. For quadrupedal locomotion ($g = 0$), the total reward is defined as $r^{\text{quad}} = r^{\text{quad}}_{\text{track}} + r^{\text{quad}}_{\text{reg}}$. For bipedal locomotion ($g = 1$), the total reward is given by $r^{\text{bip}} = r^{\text{bip}}_{\text{track}} + r^{\text{bip}}_{\text{stand}} + r^{\text{bip}}_{\text{reg}}$. The detailed reward functions can be found in Appendix V-A.

Termination: The robot terminates under different circumstances under different gait modes. When $g = 0$, the robot terminates when $\theta_{\text{roll}} > 1.0$ or $\theta_{\text{pitch}} > 1.6$. When $g = 1$, the robot terminates when any other links except rear feet and calf contacts the ground after 1 second.

Training: Our training primarily follows the Probability Annealing Selection (PAS) paradigm [54]. Overall, the robot utilizes both privileged and proprioceptive information in the first stage; but in the second stage, it learns to rely exclusively on proprioception for locomotion, using an estimator to estimate the privileged latent. In the first training stage, all observation states $[\boldsymbol{p}_t, \boldsymbol{e}_t, \boldsymbol{i}_t, \boldsymbol{c}_t]$ are accessible to train an Oracle policy. The implicit state $\boldsymbol{i}_t$ is first encoded through an encoder network into a latent representation, which is then concatenated with the explicit privileged state $\boldsymbol{e}_t$ and proprioception $\boldsymbol{p}_t$ to form the dual-state representation $\boldsymbol{l}_t = [\text{Enc}(\boldsymbol{i}_t), \boldsymbol{e}_t, \boldsymbol{p}_t]$. Then, the downstream LSTM integrates historical information into state $\boldsymbol{h}_t$. As discussed in subsection IV-C, jointly learning multiple tasks in multitask reinforcement learning (MTRL) often leads to gradient conflicts. To address this issue, we incorporate the Mixture of Experts (MoE) architecture into both the actor and critic networks, effectively mitigating gradient conflicts and improving learning efficiency. Specifically, each MoE module f operates as follows:

$$\hat{\boldsymbol{g}}_i = \text{softmax}(g(\boldsymbol{h}_t))[i], \tag{2}$$

$$\boldsymbol{a}_t = \sum_{i=1}^{N} \hat{\boldsymbol{g}}_i \cdot f_i(\boldsymbol{h}_t), \tag{3}$$

Here, $\boldsymbol{h}_t$ is the output of the low-level LSTM module, g is the gating network that outputs the gating scores, and f_i denotes expert i. Additionally, we pretrain the estimator module in this stage using a L2 loss L_{recon} to reconstruct $\text{Estimator}(\boldsymbol{p}_t, \boldsymbol{c}_t)$ into $[\text{Enc}(\boldsymbol{i}_t), \boldsymbol{e}_t]$. In summary, the overall optimization objective is:

$$L_{\text{surro}} + L_{\text{value}} + L_{\text{recon}}, \tag{4}$$

where L_{surro} and L_{value} are surrogate loss and value loss in PPO algorithm.

In the second training stage, the policy can only access $[\boldsymbol{p}_t, \boldsymbol{c}_t]$ as observations. The weights of the estimator, the low-level LSTM, and the MoE modules are initialized by copying them from the first training stage. Probability Annealing Selection [54] is then employed to gradually adapt the policy to inaccurate estimates with minimal degradation of the Oracle policy performance. Detailed pseudocode is in subsection V-B

The MoE architecture facilitates the coordination of similar task skills while minimizing conflicts between heterogeneous tasks by dynamically routing tasks to appropriate experts. This automatic routing enables specialization, improving both efficiency and task performance. Additionally, we incorporate MoE into the critic network to better capture diverse task reward structures. The actor MoE, critic MoE, and gating network share the same input h_t, which encodes proprioceptive states and task-specific features. By using a shared gating network, we ensure consistency between policy evaluation and action generation.

C. Skill Decomposition and Composition

A key challenge in multitask learning is how to efficiently utilize previous acquired locomotion skills to form new locomotion tasks. The MoE framework offers a natural solution by dynamically decompose tasks into expert of different skills. Specifically, we conduct quantitative analyses on expert

TABLE I: **Quantitative Comparison in Simulation.** Metrics include success rate, average travel distance, and average passing time.

Method	Mix	Bar (q)	Baffle (q)	Stair (q)	Pit (q)	Slope (q)	Walk (b)	Slope (b)	Stair (b)
	Success Rate ↑								
Ours	**0.879**	**0.886**	**0.924**	**0.684**	**0.902**	0.956	**0.932**	**0.961**	**0.964**
Ours w/o MoE	0.571	0.848	0.264	0.568	0.698	**0.988**	0.826	0.504	0.453
RMA	0.000	0.871	0.058	0.017	0.017	0.437	0.000	0.000	0.000
	Average Pass Time (s) ↓								
Ours	**230.98**	**102.42**	**87.84**	**179.14**	**91.86**	76.75	**92.37**	**86.14**	**86.44**
Ours w/o MoE	315.47	125.46	318.68	214.52	161.38	**65.28**	156.76	236.67	253.62
RMA	400.00	107.84	385.25	395.34	394.49	272.06	400.00	400.00	400.00
	Average Travel Distance (m) ↑								
Ours	**89.41**	**28.05**	**28.02**	20.42	**27.82**	27.62	**27.20**	**27.99**	**28.04**
Ours w/o MoE	57.12	27.59	17.41	**22.66**	25.59	**28.49**	22.73	26.21	14.23
RMA	13.40	27.39	11.31	3.92	12.48	21.33	2.00	2.00	2.00

coordination across various tasks. Detailed experiments and results are presented in subsection IV-E.

With the automatic decomposition of expert skills, we can recombine them with adjustable weights to synthesize new skills and gaits. Formally, we leverage pretrained experts and modify the gating weights as follows:

$$\hat{\boldsymbol{g}}_i = w[i] \cdot \text{softmax}(g(\boldsymbol{h}_t))[i], \tag{5}$$

where $w[i]$ can be manually defined or dynamically adjusted by a neural network. This formulation enables controlled skill blending, allowing the robot to adapt and generalize to novel locomotion patterns.

Each expert naturally specializes in different aspects of movement, such as balancing, crawling, or obstacle crossing. By selectively adjusting the contributions of these experts, we can construct new locomotion strategies without requiring additional training. This recomposition process highlights the interpretability of MoE-based policies, as each expert's role can be explicitly identified and manipulated. Detailed experiments are provided in subsection IV-F and subsection IV-G.

IV. EXPERIMENTS

A. Experiment Setup

We conduct our simulation training in IsaacGym [17], utilizing 4096 robots concurrently on an NVIDIA RTX 3090 GPU. Training begins with 40,000 iterations for plane walking in both gaits, followed by 80,000 iterations on challenging terrain tasks. Finally, we apply PAS for 10,000 iterations to adapt the policy to pure proprioception input. The control frequency in both the simulation environment and the real world is 50Hz. Our policy is deployed on the Unitree Go2 quadruped robot, with an NVIDIA Jetson Orin serving as the onboard computing device. We use PD control for low-level joint execution ($K_p = 40.0, K_d = 0.5$). We select expert number N_{exp} as 6. In all experiment results, **q** represents **quadrupedal gait**, and **b** represents **bipedal gait**.

B. Multitask Performance

1) Simulation Experiment: We conduct comparative simulation experiments with the following baselines:

- **Ours w/o MoE** [20]: Uses the same framework as ours but replaces the MoE module with a simple MLP. We control the total parameter to be the same as our MoE policy.
- **RMA** [3]: Employs a 1D-CNN as an asynchronous adaptation module within a teacher-student training framework, without using an MoE module.

We constructed a benchmark for quadrupedal robot locomotion across different tasks. Our benchmark consists of a $5m \times 100m$ runway with various obstacles evenly distributed along the path. The obstacles include:

TABLE II: Benchmark tasks for simulation experiments

Obstacle Type	Specification	Gait Mode
Bars	5 bars, height: 0.05m – 0.2m	Quadrupedal
Pits	5 pits, width: 0.05m – 0.2m	Quadrupedal
Baffles	5 baffles, height: 0.3m – 0.22m	Quadrupedal
Up Stairs	3 sets, step height: 5cm – 15cm	Quadrupedal
Down Stairs	3 sets, step height: 5cm – 15cm	Quadrupedal
Up Slopes	3 sets, incline: 10° – 35°	Quadrupedal
Down Slopes	3 sets, incline: 10° – 35°	Quadrupedal
Plane	10m flat surface	Bipedal
Up Slopes	3 sets, incline: 10° – 35°	Bipedal
Down Slopes	3 sets, incline: 10° – 35°	Bipedal
Down Stairs	3 sets, step height: 5cm – 15cm	Bipedal

We also conducted experiments for each challenging tasks, each track is 30 meters long. We consider three metrics: **Success Rate**, **Average Pass Time**, and **Average Travel Distance**. A trial is considered successful if the robot reaches within 1m of the target point within 400 seconds. Upon completion, we record the travel time. Failure cases include falling off the runway, getting stuck, or meeting the termination conditions outlined in subsection III-B. For failed trials, the pass time is recorded as 400 seconds. We compute the overall success rate and average pass time across all trials. Additionally, we measure the average lateral travel distance of all robots at the end of the evaluation.

As shown in Table I, our MoE policy achieves the best performance in the mixed-task benchmark across all three metrics. In all single-task evaluations, except for quadrupedal slope walking, our policy outperforms others. This exception may be attributed to the relative simplicity of quadrupedal

slope walking. Furthermore, policy without MoE struggles to effectively traverse the challenging multitask terrain setting, as it is significantly affected by gradient conflicts, as discussed in subsection IV-C. Regarding RMA, we adhere to its original implementation, which utilizes only an MLP backbone and a CNN encoder. This design choice leads to its suboptimal performance on multiple challenging terrains.

2) Real World Experiments: We deploy our MoE policy zero-shotly on real robots and conduct real world experiments. We test mix terrain that contains all challenging tasks, as well as each separated terrains. For the mix terrain, our robot first need to consequently cross 20cm bar, 22cm baffle, 15cm stairs, 20cm pits and 30 degree slopes in a quadrupedal gait. Then it receives a bipedal command to stand up, walk up the 30 degree slope, turn around and walk down. For each tests, we test for 20 trails and record the average success rate. We also test different policies for single tasks.

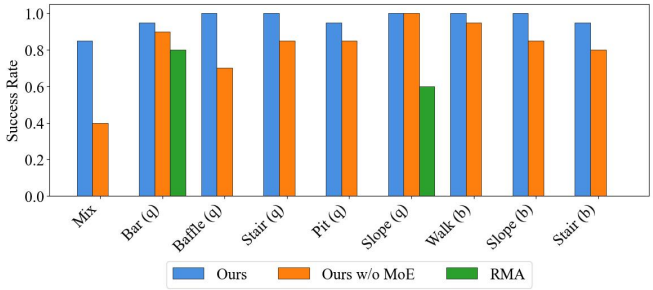

Fig. 4: **Real world success rate over multiple terrains and gaits.**

As shown in Figure 4, our method achieves better performance across all types of tasks and demonstrates a significantly higher success rate in mix terrain. Additionally, we conduct experiments in more outdoor environments, as shown in Figure 5, further demonstrating its robustness and generalization.

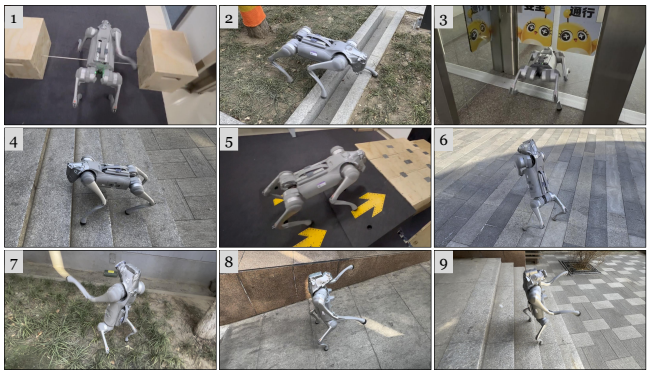

Fig. 5: **Real-world experiments over multiple terrains and gaits:** 1. Bar (Quad), 2. Pit (Quad), 3. Baffle (Quad), 4. Stair (Quad), 5. Slope (Quad), 6. Stand up (Bip), 7. Walk (Bip), 8. Slope (Bip), 9. Stair (Bip).

C. Gradient Conflict Alleviation

In order to unveil whether gradient conflict can be reduced by applying mixture of experts, we conducted gradient conflict experiments. We resume from the checkpoints after pretraining for 15000 epochs and run the training process of multitask for 500 epochs of 4096 quadrupedal robots. We average the gradient throughout the process. We consider two metrics to measure the gradient conflict of robot locomotion. **1) Cosine Similarity:** We compute the normalized dot product of all parameters' gradients of different tasks. Smaller cosine similarity indicates larger gradient conflicts. **2) Negative gradient ratio:** We compute negative gradient update ratio for each pair of task gradients. Larger negative gradient ratio indicates larger gradient conflict. We test the gradient conflict between 5 different tasks: quadrupedal bar crossing, quadrupedal baffle crawling, quadrupedal stair walking, bipedal plane walking and bipedal slope walking.

TABLE III: **Cosine similarity of gradients of different tasks.** Left represents MoE policy and right represents standard policy.

MoE/Standard	Gradient Cosine Similarity ↑				
	Bar (q)	**Baffle (q)**	**Stair (q)**	**Slope Up (b)**	**Slope down (b)**
Bar (q)	-	**0.519**/0.474	**0.606**/0.592	**0.278**/-0.132	**0.091**/-0.128
Baffle (q)	-	-	0.369/**0.384**	**0.062**/-0.091	**0.061**/-0.101
Stair (q)	-	-	-	**0.046**/-0.023	**0.052**/0.015
Slope up (b)	-	-	-	-	**0.806**/0.709
Slope down (b)	-	-	-	-	-

TABLE IV: **Negative entry ratio of MoE and Standard policy on different tasks.** Left represents MoE policy and right represents standard policy.

MoE/Standard	Gradient Negative Entries (%) ↓				
	Bar (q)	**Baffle (q)**	**Stair (q)**	**Slope Up (b)**	**Slope down (b)**
Bar (q)	-	**35.72**/37.33	32.67/**32.62**	**45.50**/ 55.12	**49.83**/50.80
Baffle (q)	-	-	39.90/**38.52**	**49.86**/55.91	**49.91**/51.68
Stair (q)	-	-	-	**49.52**/50.15	**50.04**/50.34
Slope up (b)	-	-	-	-	**23.17**/30.91
Slope down (b)	-	-	-	-	-

As shown in Table III and Table IV, the MoE policy significantly reduces gradient conflict between bipedal and quadrupedal tasks. It also minimizes gradient conflict even between quadrupedal tasks that require fundamentally different skills, such as quadrupedal bar crossing and quadrupedal baffle crawling.

D. Training Performance

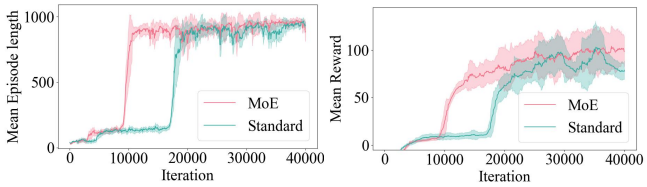

Fig. 6: **Training curve of our multitask policy in the pretraining stage.**

In terms of training performance, we focus on the mean reward and mean episode length. The mean reward reflects the policy's ability to exploit the environment, while the mean episode length indicates how well the robot learns to stand and walk. We present the training curve during the plane pretraining stage, where the robot learns both bipedal and quadrupedal plane walking. As shown in Figure 6, our MoE policy outperforms the standard policy with similar total parameters across both metrics.

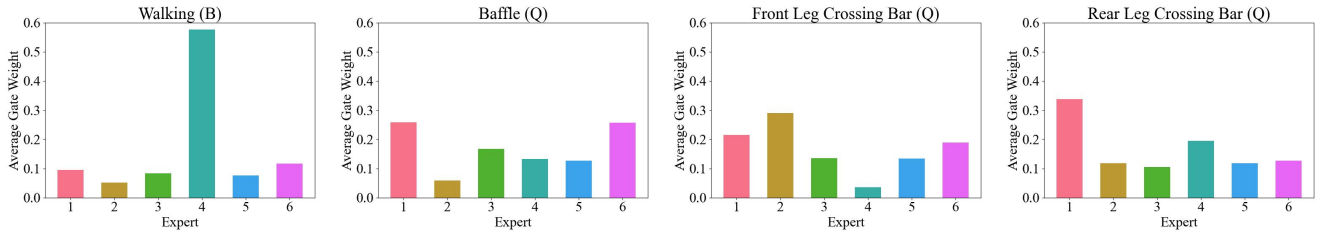

Fig. 7: **Expert usage in different tasks.** From left to right is: Bipedal walking, Quadrupedal Baffle crawling, Front Leg Crossing Bars and Rear Leg Crossing Bars.

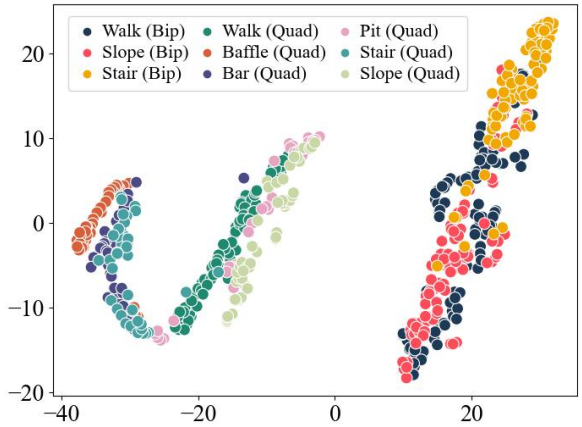

Fig. 8: **t-SNE result of gating network output on different terrains and gaits.**

E. Expert Specialization Analysis

We conduct both qualitative and quantitative experiments to analyze the emergent composition of skills. As shown in Figure 7, we plot the mean weight of different experts across various tasks. It is evident that the distribution of gating weights varies significantly from task to task, demonstrating the expertise and differentiation of various experts.

To further analyze the composition of different experts across various tasks, we use t-SNE to visualize the output of the gating network for different tasks (i.e., the weight of each expert). As shown in Figure 8, the bipedal and quadrupedal tasks form distinct clusters. Quadrupedal slope walking and quadrupedal pit crossing tasks are performed using a gait similar to quadrupedal plane walking, and thus, they cluster closely together. In contrast, bar crossing, baffle crawling, and stair climbing exhibit a gait that differs more significantly, resulting in them clustering further apart.

F. Skill Composition

Fig. 9: **Manually designed new dribbling gait by selecting two experts.**

During our training, we found that our experts not only specialize and cooperate across different tasks, but they also emerge with specific, human-interpretable skills. We discovered that one expert specializes in balancing, which helps lift the robot's body but limits its agility. Another expert is responsible for lifting one of the front legs to perform crossing tasks, enabling the robot to execute basic movement skills. By selecting the balancing expert and the crossing expert, we are able to zero-shot transfer to a new dribbling pattern. In this gait, we select the two experts mentioned above, manually double the gating weight of the crossing expert, and mask out all other experts. As shown in Figure 9, the new dribbling gait allows the robot to walk effectively while periodically using one of its front legs to kick the ball. This skill composition results from the automatic skill decomposition and interpretability inherent in the MoE architecture. In contrast, a standard neural network would function as a black box, lacking such interpretable skill decomposition.

G. Additional Experiment

Fig. 10: **MoE-Loco can quickly adapt to a three-footed gait by training a new expert.** 1) ground plane, 2) slope up, and 3) slope down.

We conduct an adaptation learning experiment to demonstrate how our pretrained experts can be recomposed and adapted to new tasks. In this experiment, we design the robot to walk on three feet. We introduce a newly initialized expert, freeze the parameters of the original experts, and update only the gating network. As shown in Figure 10, the robot can walk on both flat ground and a slope using only three feet. The newly added expert only needs to learn how to lift one leg, while leveraging the walking and slope capabilities of the original experts.

V. CONCLUSION

We introduced MoE-Loco, a multitask locomotion framework that utilizes a Mixture of Experts architecture to train a single policy for quadrupedal robots. Our approach effectively mitigates gradient conflicts, leading to improved training efficiency and task performance. Through extensive evaluations in both simulated and real-world environments, we demonstrated the capability of our method to handle a variety of terrains and gaits. Future work will explore extending this approach to incorporate sensory perception such as camera and Lidar to enhance adaptability in more complex tasks.

APPENDIX

A. Reward Functions

The robots in bipedal gait receives different reward to the robots in quadrupedal gait. Detailed explanation is shown in Table V.

TABLE V: Reward functions

Type	Item	Formula	Weight
Quadrupedal Tracking	Tracking lin vel	$\exp\left(-\frac{\lVert \mathbf{v}_{c,xy}-\mathbf{v}_{b,xy}\rVert^2}{\sigma}\right)$	7.0
	Tracking ang vel	$\exp\left(-\frac{(c_{yaw}-\omega_{yaw})^2}{\sigma}\right)$	2.5
	Termination	-1	-1.0
	Alive	1	1.0
Quadrupedal Regularization	Joint pos	$(\mathbf{q}-\mathbf{q}_{\text{default}})^2$	-0.05
	Joint vel	$\lVert\dot{\mathbf{q}}\rVert_2$	-0.002
	Joint acc	$\lVert\ddot{\mathbf{q}}\rVert_2$	-2×10^{-6}
	Ang vel stability	$\left(\lVert\omega_{t,x}\rVert_2+\lVert\omega_{t,y}\rVert_2\right)$	-0.2
	Feet in air	$\prod_{i=1}^{\mathbf{N}}\mathbb{I}\left\{\mathbf{F}_{i,z}^{\text{foot}}<1\right\}+\sum_{i=1}^{\mathbf{N}}\mathbb{I}\left\{\mathbf{F}_{i,z}^{\text{foot}}<1\right\}\mathbb{I}\left\{\mathbf{F}_{i,z}^{\text{calf}}\geq1\right\}$	-0.05
	Front hip pos	$\sum_{i\in I_{\text{front hip}}}\left(\mathbf{q}_i-\mathbf{q}_i^{\text{default}}\right)^2$	-0.2
	Rear hip pos	$\sum_{i\in I_{\text{rear hip}}}\left(\mathbf{q}_i-\mathbf{q}_i^{\text{default}}\right)^2$	-0.5
	Base height	$\left(\mathbf{z}_{\text{base}}-\frac{1}{\mathbf{N}}\sum_{i=1}^{\mathbf{N}}\mathbf{h}_i-\mathbf{h}_{\text{target}}\right)^2$	-0.1
	Balance	$\lVert\mathbf{F}_{\text{feet},0}+\mathbf{F}_{\text{feet},2}-\mathbf{F}_{\text{feet},1}-\mathbf{F}_{\text{feet},3}\rVert_2$	-2×10^{-5}
	Joint limit	$\left\lVert\left(\mathbf{q}<\mathbf{q}_{\min}\right)\vee\left(\mathbf{q}>\mathbf{q}_{\max}\right)\right\rVert_1$	-0.01
	Torque exceed limit	$\sum_{i=1}^{\mathbf{N}_{\text{sub}}}\sum_{j=1}^{12}\max\left\{\lvert\tau_{ij}\rvert-\tau_j^{\text{limit}},0\right\}$	-2.0
Bipedal Stand	Orientation	$\left(0.5\cos\theta+0.5\right)^2,\quad\theta=\arccos\left(\frac{\mathbf{g}\cdot\mathbf{i}}{\lVert\mathbf{g}\rVert\lVert\mathbf{i}\rVert}\right)$	1.0
	Base height linear	$\min\left(\max\left(\frac{z_{\text{base}}-T_{\min}}{T_{\max}-T_{\min}},0\right),1\right)$	0.8
Bipedal Tracking	Tracking lin vel	$\exp\left(-\frac{\lVert\mathbf{v}_c-\mathbf{v}_b\rVert^2}{\sigma}\right)\mathbb{I}\left\{\cos\theta>0.95\right\}\frac{z_{\text{base}}-s_{\text{low}}}{s_{\text{high}}-s_{\text{low}}}$	3.0
	Tracking ang vel	$\exp\left(-\frac{(\omega_{\text{roll}}-\omega_{\text{roll}})^2}{\sigma_{ang}}\right)\mathbb{I}\left\{\cos\theta>0.95\right\}\frac{z_{\text{base}}-s_{\text{low}}}{s_{\text{high}}-s_{\text{low}}}$	2.5
	Termination	-1	-1.0
	Alive	1	1.0
Bipedal Regularization	Rear air	$\prod_{i\in\mathbf{R}}\mathbb{I}\left\{\mathbf{f}_{i,z}^{\text{rear}}<1\right\}+\sum_{i\in\mathbf{R}}\mathbb{I}\left\{\mathbf{f}_{i,z}^{\text{rear}}<1\right\}\mathbb{I}\left\{\mathbf{f}_{i,z}^{\text{rear-calf}}\geq1\right\}$	-0.5
	Front hip pos	$\sum_{i\in\mathbf{F}_H}\left(\mathbf{q}_i-\mathbf{q}_i^{\text{default}}\right)^2$	-0.1
	Rear hip pos	$\sum_{i\in\mathbf{R}_H}\left(\mathbf{q}_i-\mathbf{q}_i^{\text{default}}\right)^2$	-0.18
	Rear pos balance	$\left\lVert\mathbf{q}_i^{\text{rear left}}-\mathbf{q}_i^{\text{rear right}}\right\rVert_2$	-0.05
	Front joint pos	$\mathbf{1}\left\{\mathbf{t}>\mathbf{T}_{\text{allow}}\right\}\sum_{i\in\mathbf{F}}\left(\mathbf{q}_i-\mathbf{q}_i^{\text{default}}\right)^2$	-0.2
	Front joint vel	$\mathbf{1}\left\{\mathbf{t}>\mathbf{T}_{\text{allow}}\right\}\sum_{i\in\mathbf{F}}\dot{\mathbf{q}}_i^2$	-1×10^{-3}
	Front joint acc	$\mathbf{1}\left\{\mathbf{t}>\mathbf{T}_{\text{allow}}\right\}\sum_{i\in\mathbf{F}}\left(\frac{\Delta\dot{\mathbf{q}}_i}{\Delta t}\right)^2$	-2×10^{-6}
	Legs energy substeps	$\frac{1}{\mathbf{N}_s}\sum_{j=1}^{\mathbf{N}_s}\sum_{i=1}^{\mathbf{N}_{\text{dof}}}\left(\tau_{ij}\dot{\mathbf{q}}_{ij}\right)^2$	-1×10^{-6}
	Torque exceed limits	$\sum_j\sum_i\max\left\{\lvert\tau_{ij}\rvert-\tau_i^{\text{limit}},0\right\}$	-2.0
	Joint limits	$\frac{1}{\mathbf{N}_s}\sum_j\sum_i\mathbb{I}\left\{\mathbf{q}_{ij}\notin[\mathbf{q}_i^{\min},\mathbf{q}_i^{\max}]\right\}$	-0.06
	Collision	$\sum_k\mathbb{I}\left\{\lVert\mathbf{f}_k\rVert>1\right\}$	-2.0
	Action rate	$\lVert\mathbf{a}_i^{\text{last}}-\mathbf{a}_i\rVert_2$	-0.03
	Joint vel	$\lVert\dot{\mathbf{q}}\rVert_2$	-2×10^{-3}
	Joint acc	$\lVert\ddot{\mathbf{q}}\rVert_2$	-3×10^{-6}

B. Algorithm Pseudocode

Algorithm 1 Training Stage 1

for total iteration **do**
  Initialize rollout buffer $\mathcal{D}\leftarrow\emptyset$
  for num steps **do**
    $\mathbf{z}_t\leftarrow\text{Enc}(\mathbf{i}_t)$
    $\hat{\mathbf{l}}_t\leftarrow[\text{Estimator}(\mathbf{p}_t,\mathbf{c}_t),\mathbf{p}_t]$
    $\mathbf{l}_t\leftarrow[\mathbf{z}_t,\mathbf{e}_t,\mathbf{p}_t]$
    $\mathbf{h}_t\leftarrow\text{LSTM}([\mathbf{l}_t,\mathbf{c}_t])$
    $\hat{\mathbf{g}}\leftarrow\text{softmax}(g(\mathbf{h}_t))$
    $\mathbf{a}_t\leftarrow\sum_{i=1}^{N}\hat{\mathbf{g}}_i\cdot f_i(\mathbf{h}_t)$
    Execute $\mathbf{a}_t$, observe reward r_t and next state
    Reset if terminated
    $t\leftarrow t+1$
    Store rollout in $\mathcal{D}$
  end for
  Compute PPO losses: L_{surro}, L_{value}
  Compute reconstruction loss:
  $L_{\text{recon}}=\sum_{\hat{\mathbf{l}}_i,\mathbf{l}_i\in\mathcal{D}}\left\lVert\hat{\mathbf{l}}_i-\mathbf{l}_i\right\rVert^2$
  $L=L_{\text{surro}}+L_{\text{value}}+L_{recon}$
  Update policy and value network
end for
return Oracle Policy

Algorithm 2 Training Stage 2

Copy parameters from Oracle Policy
for total iteration **do**
  Initialize rollout buffer $\mathcal{D}\leftarrow\emptyset$
  for num steps **do**
    $\hat{\mathbf{l}}_t\leftarrow[\text{Estimator}(\mathbf{p}_t,\mathbf{c}_t),\mathbf{p}_t]$
    $\mathbf{l}_t\leftarrow[\mathbf{z}_t,\mathbf{e}_t,\mathbf{p}_t]$
    $\mathbf{P}_t\leftarrow\alpha^t$
    $\bar{\mathbf{l}}_t\leftarrow\text{Probability Selection}(\mathbf{P}_t,\hat{\mathbf{l}}_t,\mathbf{l}_t)$
    $\mathbf{h}_t\leftarrow\text{LSTM}([\bar{\mathbf{l}}_t,\mathbf{c}_t])$
    $\hat{\mathbf{g}}\leftarrow\text{softmax}(g(\mathbf{h}_t))$
    $\mathbf{a}_t\leftarrow\sum_{i=1}^{N}\hat{\mathbf{g}}_i\cdot f_i(\mathbf{h}_t)$
    Execute $\mathbf{a}_t$, observe reward r_t and next state
    Reset if terminated
    $t\leftarrow t+1$
    Store rollout in $\mathcal{D}$
  end for
  Compute PPO losses: L_{surro}, L_{value}
  Compute reconstruction loss:
  $L_{\text{recon}}=\sum_{\hat{\mathbf{l}}_i,\mathbf{l}_i\in\mathcal{D}}\left\lVert\hat{\mathbf{l}}_i-\mathbf{l}_i\right\rVert^2$
  $L=L_{\text{surro}}+L_{\text{value}}+L_{recon}$
  Update policy and value network
end for
return Final Policy

C. Network Architecture Details

The architecture of different modules used in our experiment is shown in Table VI.

TABLE VI: Network architecture details

Network	Type	Hidden dims
Actor RNN	LSTM	[256]
Critic RNN	LSTM	[256]
Estimator Module	LSTM	[256]
Estimator Latent Encoder	MLP	[256, 128]
Implicit Encoder	MLP	[32, 16]
Expert Head	MLP	[256, 128, 128]
Standard Head	MLP	[640, 384, 128]
Gating Network	MLP	[128]

REFERENCES

[1] K. N. Kumar, I. Essa, and S. Ha, "Cascaded compositional residual learning for complex interactive behaviors," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4601–4608, 2023. 1, 2

[2] A. Klipfel, N. Sontakke, R. Liu, and S. Ha, "Learning a single policy for diverse behaviors on a quadrupedal robot using scalable motion imitation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 2768–2775. 1

[3] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021. 1, 2, 4

[4] Z. Su, X. Huang, D. Ordoñez-Apraez, Y. Li, Z. Li, Q. Liao, G. Turrisi, M. Pontil, C. Semini, Y. Wu, *et al.*, "Leveraging symmetry in rl-based legged locomotion control," *arXiv preprint arXiv:2403.17320*, 2024. 1, 2

[5] G. Ji, J. Mun, H. Kim, and J. Hwangbo, "Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4630–4637, 2022. 1

[6] J. Long, J. Ren, M. Shi, Z. Wang, T. Huang, P. Luo, and J. Pang, "Learning humanoid locomotion with perceptive internal model," *arXiv preprint arXiv:2411.14386*, 2024. 1

[7] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao, "Robot parkour learning," *arXiv preprint arXiv:2309.05665*, 2023. 1, 2

[8] J. Wu, G. Xin, C. Qi, and Y. Xue, "Learning robust and agile legged locomotion using adversarial motion priors," *IEEE Robotics and Automation Letters*, 2023. 1, 2

[9] S. Luo, S. Li, R. Yu, Z. Wang, J. Wu, and Q. Zhu, "Pie: Parkour with implicit-explicit learning framework for legged robots," *arXiv preprint arXiv:2408.13740*, 2024. 1, 2

[10] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 11 443–11 450. 1, 2

[11] S. Liu, Z. Chen, Y. Liu, Y. Wang, D. Yang, Z. Zhao, Z. Zhou, X. Yi, W. Li, W. Zhang, *et al.*, "Improving generalization in visual reinforcement learning via conflict-aware gradient agreement augmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 23 436–23 446. 1

[12] S. Zhou, W. Zhang, J. Jiang, W. Zhong, J. Gu, and W. Zhu, "On the convergence of stochastic multi-objective gradient manipulation and beyond," *Advances in Neural Information Processing Systems*, vol. 35, pp. 38 103–38 115, 2022. 1

[13] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991. 1, 2

[14] J. Obando-Ceron, G. Sokar, T. Willi, C. Lyle, J. Farebrother, J. Foerster, G. K. Dziugaite, D. Precup, and P. S. Castro, "Mixtures of experts unlock parameter scaling for deep rl," *arXiv preprint arXiv:2402.08609*, 2024. 1, 2

[15] K. Li, M. Cucuringu, L. Sánchez-Betancourt, and T. Willi, "Mixtures of experts for scaling up neural networks in order execution," in *Proceedings of the 5th ACM International Conference on AI in Finance*, 2024, pp. 669–676. 1

[16] O. Celik, A. Taranovic, and G. Neumann, "Acquiring diverse skills using curriculum reinforcement learning with mixture of experts," *arXiv preprint arXiv:2403.06966*, 2024. 1, 2

[17] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021. 2, 4

[18] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023. 2

[19] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019. 2

[20] S. Zhu, R. Huang, L. Mou, and H. Zhao, "Robust robot walker: Learning agile locomotion over tiny traps," *arXiv preprint arXiv:2409.07409*, 2024. 2, 3, 4

[21] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020. 2

[22] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid locomotion via reinforcement learning," *The International Journal of Robotics Research*, vol. 43, no. 4, pp. 572–587, 2024. 2

[23] T. He, C. Zhang, W. Xiao, G. He, C. Liu, and G. Shi, "Agile but safe: Learning collision-free high-speed legged locomotion," *arXiv preprint arXiv:2401.17583*, 2024. 2

[24] Y. Li, J. Li, W. Fu, and Y. Wu, "Learning agile bipedal motions on a quadrupedal robot," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 9735–9742. 2

[25] L. Smith, J. C. Kew, T. Li, L. Luu, X. B. Peng, S. Ha, J. Tan, and S. Levine, "Learning and adapting agile locomotion skills by transferring experience," *arXiv preprint arXiv:2304.09834*, 2023. 2

[26] Y. Cheng, H. Liu, G. Pan, L. Ye, H. Liu, and B. Liang, "Quadruped robot traversing 3d complex environments with limited perception," *arXiv preprint arXiv:2404.18225*, 2024. 2

[27] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "Anymal parkour: Learning agile navigation for quadrupedal robots," *Science Robotics*, vol. 9, no. 88, p. eadi7566, 2024. 2

[28] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, pp. 41–75, 1997. 2

[29] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 160–167. 2

[30] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool, "Multi-task learning for dense prediction tasks: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3614–3633, 2021. 2

[31] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," *arXiv preprint arXiv:1605.05101*, 2016. 2

[32] L. Pinto and A. Gupta, "Learning to push by grasping: Using multiple tasks for effective learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 2161–2168. 2

[33] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5824–5836, 2020. 2

[34] B. Liu, X. Liu, X. Jin, P. Stone, and Q. Liu, "Conflict-averse gradient descent for multi-task learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 878–18 890, 2021. 2

[35] S. Huang, Z. Zhang, T. Liang, Y. Xu, Z. Kou, C. Lu, G. Xu, Z. Xue, and H. Xu, "Mentor: Mixture-of-experts network with task-oriented perturbation for visual reinforcement learning," *arXiv preprint arXiv:2410.14972*, 2024. 2

[36] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135. 2

[37] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel, "Rl^2: Fast reinforcement learning via slow reinforcement learning," *arXiv preprint arXiv:1611.02779*, 2016. 2

[38] S. Sodhani, A. Zhang, and J. Pineau, "Multi-task reinforcement learning with context-based representations," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9767–9779. 2

[39] C. Yang, K. Yuan, Q. Zhu, W. Yu, and Z. Li, "Multi-expert learning of adaptive legged locomotion," *Science Robotics*, vol. 5, no. 49, p. eabb2174, 2020. 2

[40] Y. Ze, G. Yan, Y.-H. Wu, A. Macaluso, Y. Ge, J. Ye, N. Hansen, L. E. Li, and X. Wang, "Gnfactor: Multi-task real robot learning with generalizable neural feature fields," in *Conference on Robot Learning*. PMLR, 2023, pp. 284–301. 2

[41] M. Shafiee, G. Bellegarda, and A. Ijspeert, "Manyquadrupeds: Learning a single locomotion policy for diverse quadruped robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 3471–3477. 2

[42] C. Yang, K. Yuan, Q. Zhu, W. Yu, and Z. Li, "Multi-expert learning of adaptive legged locomotion," *Science Robotics*, vol. 5, no. 49, p. eabb2174, 2020. 2

[43] N. Shah, K. Tiwari, and A. Bera, "Mtac: Hierarchical reinforcement learning-based multi-gait terrain-adaptive quadruped controller," *arXiv preprint arXiv:2401.03337*, 2023. 2

[44] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the em algorithm," *Neural computation*, vol. 6, no. 2, pp. 181–214, 1994. 2

[45] M. Deisenroth and J. W. Ng, "Distributed gaussian processes," in *International conference on machine learning*. PMLR, 2015, pp. 1481–1490. 2

[46] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "Gshard: Scaling giant models with conditional computation and automatic sharding," *arXiv preprint arXiv:2006.16668*, 2020. 2

[47] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand, *et al.*, "Mixtral of experts," *arXiv preprint arXiv:2401.04088*, 2024. 2

[48] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, *et al.*, "Deepseek-v3 technical report," *arXiv preprint arXiv:2412.19437*, 2024. 2

[49] Z. Zhang, S. Liu, J. Yu, Q. Cai, X. Zhao, C. Zhang, Z. Liu, Q. Liu, H. Zhao, L. Hu, *et al.*, "M3oe: Multi-domain multi-task mixture-of experts recommendation framework," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 893–902. 2

[50] J. Jiang, P. Zhang, Y. Luo, C. Li, J. B. Kim, K. Zhang, S. Wang, X. Xie, and S. Kim, "Adamct: adaptive mixture of cnn-transformer for sequential recommendation," in *Proceedings of the 32nd ACM international conference on information and knowledge management*, 2023, pp. 976–986. 2

[51] Y. Gou, Z. Liu, K. Chen, L. Hong, H. Xu, A. Li, D.-Y. Yeung, J. T. Kwok, and Y. Zhang, "Mixture of cluster-conditional lora experts for vision-language instruction tuning," *arXiv preprint arXiv:2312.12379*, 2023. 2

[52] S. Chen, Z. Jie, and L. Ma, "Llava-mole: Sparse mixture of lora experts for mitigating data conflicts in instruction finetuning mllms," *arXiv preprint arXiv:2401.16160*, 2024. 2

[53] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017. 3

[54] S. Zhu, D. Li, L. Mou, Y. Liu, N. Xu, and H. Zhao, "Saro: Space-aware robot system for terrain crossing via vision-language model," *arXiv preprint arXiv:2407.16412*, 2024. 3