2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

GeRM: A Generalist Robotic Model with Mixture-of-experts for Quadruped Robot

Wenxuan Song[†,1,2], Han Zhao[†,1], Pengxiang Ding[1], Can Cui[1], Shangke Lyu[1], Yaning Fan[1], Donglin Wang[1,*]
[1]MiLAB, Westlake University, China
[2]AIM Lab, Faculty of IT, Monash University, Australia

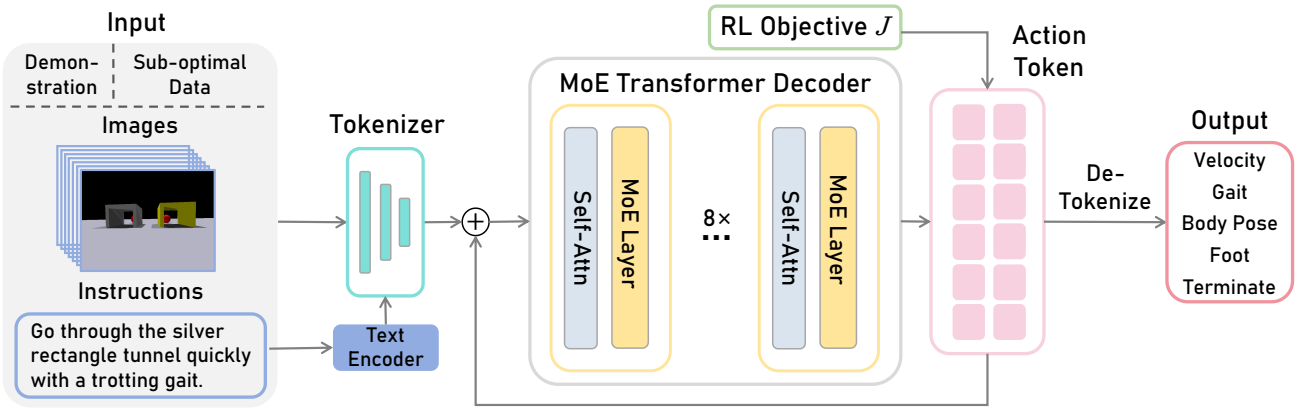

Fig. 1: **Overview of GeRM.** We take both demonstration and sub-optimal data as input. Then the images and instructions are tokenized and sent into the mixture-of-experts Transformer Decoder to generate action tokens. They are finally de-tokenized into discretized robot commands. The actions are used for RL objectives when training.

***Abstract*— Multi-task robot learning holds significant importance in tackling diverse and complex scenarios. However, current approaches are hindered by performance issues and difficulties in collecting training datasets. In this paper, we propose GeRM (Generalist Robotic Model). We utilize offline reinforcement learning to optimize data utilization strategies to learn from both demonstrations and sub-optimal data, thus surpassing the limitations of human demonstrations. Thereafter, we employ a transformer-based VLA network to process multi-modal inputs and output actions. By introducing the Mixture-of-Experts structure, GeRM allows faster inference speed with higher whole model capacity, and thus resolves the issue of limited RL parameters, enhancing model performance in multi-task learning while controlling computational costs. Through a series of experiments, we demonstrate that GeRM outperforms other methods across all tasks, while also validating its efficiency in both training and inference processes. Additionally, we uncover its potential to acquire emergent skills. Additionally, we contribute the QUARD-Auto dataset, collected automatically to support our training approach and foster advancements in multi-task quadruped robot learning. This work presents a new paradigm for reducing the cost of collecting robot data and driving progress in the multi-task learning community.**

You can reach our project and video through the link: https://songwxuan.github.io/GeRM/ .

I. Introduction

Quadruped robots, known for their exceptional ability to traverse complex terrains and execute agile movements, have become a focal point in robotics research [1], [2]. Researchers have extensively utilized these robots to tackle various tasks, including autonomous navigation (e.g. urban navigation [3]), locomotion [4], [5], [6], manipulation [7], and also multi-task learning [8], [9].

To achieve the capability to handle multi-task scenarios, quadruped robots should have the ability to receive human instructions, perceive the environment, autonomously make plans, and take action. Therefore, we want to combine language and visual inputs and output actions by utilizing the Vision-Language-Action (VLA) model proposed in RT-1 [10] into quadruped robot learning.

However, the existing VLA models, which rely on expert data collected for **Imitation Learning (IL)**, have the following problems:

1. The cost of manually collecting datasets is high. IL training relies on large-scale robot datasets [11]. Current methods for collecting robot data are based on real-world environment [12], [13], which requires experts' remote control, and simulation environment, which requires environment setup and algorithm design. Meanwhile, as the robot with the most degrees of freedom (DoFs), the difficulty in controlling quadruped robots is also notably high. These factors contribute to the increased difficulty and cost associated with collecting high-quality expert quadruped data. Therefore, we hope to automatically collect datasets and utilize them for training.

2. The performance of the IL policy is limited by the degree to which experts can provide high-quality demonstrations. This paper aims to employ **Reinforcement Learning (RL)** methods to learn from auto-collected datasets and reasonably utilize sub-optimal data to break through the demonstration. To utilize pre-collected large-scaled datasets,

†: Equal contribution
*: Corresponding author

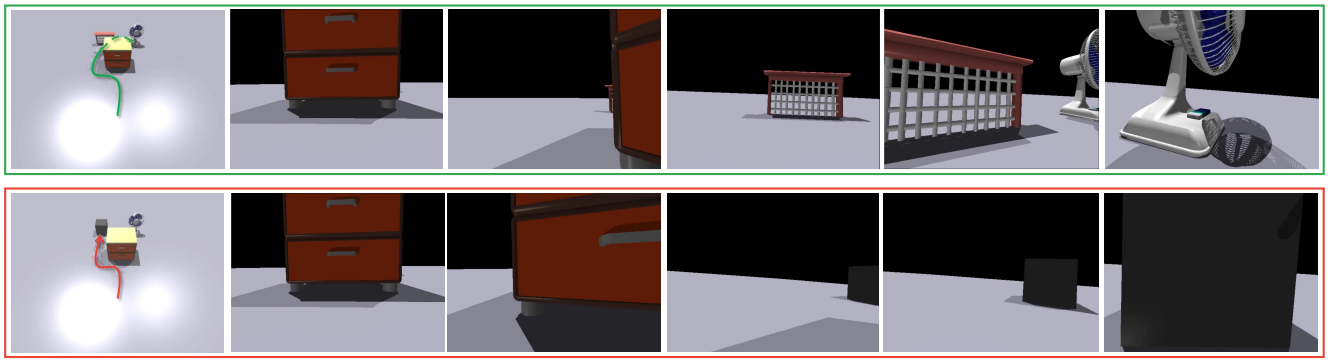

Fig. 2: **Emergent Skills.** The example of the emergent skill of dynamic adaptive path planning. We study these challenging scenarios in detail in Section V-B.

we choose the offline RL algorithm. Then the core issue is how to effectively apply the transformer-based VLA model to offline RL. Effective offline RL generally employs Deep Q-Learning. Therefore, we adopt designs akin to Q-Transformer [14] by employing a transformer-based VLA model to replace the value function and output discretized actions.

The augmentation of parameter quantity frequently enhances a model's capacity for generalization across multi-tasks, which has been proved in many fields [15], [16], [17]. However, augmenting the parameter count of an RL policy often negatively impacts its overall performance. Recently, [18] has proved the effectiveness of mixture-of-experts (MoE) to unlock parameter scaling in deep RL. Thus, we construct a mixture-of-experts structure.

GeRM is a sparse MoE network [19], [20]. It is a transformer decoder-only model where the Feed-Forward Network (FFN) picks from a set of **8** distinct groups of parameters. At every layer, for every token, a router network chooses two of these groups (the "experts") to process the token and combine their output additively. Different experts are proficient in different tasks/different action dimensions to solve problems in different scenarios, learning a generalist model across multiple tasks. This technique increases the network parameter volume while keeping the computational cost basically unchanged, as the model only uses a fraction of the total set of parameters per token.

We collected the QUARD-Auto dataset in an automatic collection manner as a supplement to our previously published QUARD dataset [21], addressing the shortcomings of sub-optimal data. It must be emphasized that we have explored a fully automated approach to data collection, which circumvents the difficulties and costs associated with manually controlling robots for demonstrations. We simply provide instructions and utilize the pre-trained VLA model to autonomously control the robot, thereafter recording both the received image and the executable action, resulting in the collection of **258418** trajectories on Issac Gym, comprising **120128** success and **138290** failures. This presents a new paradigm for the autonomous collection of large-scale robot datasets.

Our contributions mainly lie in two aspects:

- We first propose a Mixture-of-Experts model for quadruped reinforcement learning. We have adopted a Mixture-of-Experts structure to replace the conventional linear layer within the Transformer decoder, which allows faster inference speed with higher whole model capacity. Additionally, deep Q-learning methodology aims to acquire and optimize the model's capabilities to its optimal potential.
- We have extensively validated the effectiveness of GeRM through numerous experiments. It has been trained on limited demonstrations and sub-optimal data, then extensively tested across **99** tasks. GeRM outperforms existing methods and exhibits superior capabilities across multi-tasks, with only **1/2** total parameters activated. Furthermore, other experiments also demonstrate GeRM's superiority in data utilization and emergent skill development.
- We contributed an auto-collected dataset including sub-optimal data that can be used for reinforcement learning, enabling learning on sub-optimal data, thus breaking through the limitations of human demonstration data.

II. Related Work

Offline RL for Legged Robot Control. Recent works have extensively explored offline RL. [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], with Conservative Q-learning (CQL) [32] focusing on learning policies that adhere to a conservative lower bound of the value function. The objective of our research is to create an offline RL framework capable of seamless integration with high-capacity Transformers and scalable for multi-task robotic learning. Q-Transformer [14] developed a variant of CQL specifically optimized for training large Transformer-based Q-functions on mixed-quality data. Our work focuses on developing more effective and efficient strategies within the context of this specific framework.

Sparse Mixture-of-Experts Architecture. Sparse Mixture-of-Experts models have shown significant advantages in natural language processing (NLP). [33] showed that they could effectively use a very large number of weights while only activating a small subset of the computation graph when inference, which explains the term

TABLE I: **Illustration of tasks.** The "Skill" means different skill/task categories. The "Episode" signifies the number of experiments conducted for each task, which also corresponds to the number of trajectories. The "Description" is the description of the tasks. The "Example Instruction" describes different task scenarios, including various higher-level variables associated with the simulation.

Skill	Episode	Description	Example Instruction
Go to *Object*	66K	Navigate to the object and stop in front of it	Go to the trashcan slowly with a trotting gait.
Go to *Object* and avoid the obstacle	47K	Navigate to the object without colliding with the obstacle	Go to the piano and avoid the obstacle quickly with a bounding gait.
Stop *Object*	51K	Move to block the ball rolling toward the robot	Stop the red ball normally with a pacing gait.
Distinguish *Letter*	16K	Identify the correct one from multiple boxes with different printed letters	Distinguish letter B normally with a bounding gait.
Go through *Tunnel*	77K	Go through the correct tunnel from two tunnels with different colors and shapes	Go through the silver rectangle tunnel quickly with a trotting gait.
Total	257K	The total number of episodes	

"sparse". There has also been work on scaling sparse MoE architecture[34] and apply it on Transformers[35] [36]. Within it, [37] have expanded the MoE model capacity to 1 trillion parameters. Recently, in the era of LLM, MoE has become a broad and effective structure [38] [39]. MoE has also helped deep RL with parameter scalability [18]. Now we aim to apply MoE on robotic control to obtain a generalist model.

Transformer-based Vision-Language-Action Model. VLA models ([40], [41], [42], [10], [16], [43], [44], [45], [46]) integrates visual information and instructions to generate executable actions. Transformer-based VLA models hold the potential to handle general tasks by processing general inputs and outputs. Our previous work [21] has pioneered the deployment of the VLA model on quadruped robots. While existing VLA models are typically trained using imitation learning approaches, Q-Transformer [14] was the first to employ RL methods for training VLA models. We intend to further enhance the training of VLA models for quadruped robots using RL in a more effective manner.

III. Preliminaries

In RL, for a Markov decision process (MDP), there is a state s, actions a, discount factor $\gamma \in (0,1]$, transition function $T(s'|s,a)$, a reward function $R(s,a)$, and the policy π aims to maximize the total reward. Actions a have dimensionality $d_{\mathscr{A}}$. Value-based RL approaches learn a Q-function $Q(s,a)$ representing the total discounted return $\sum_t \gamma^t R(s_t,a_t)$, with policy $\pi(a|s) = \text{argmax}_a Q(s,a)$. The Q-function can be learned by iteratively applying the Bellman operator:

$$\mathscr{B}^* Q(s_t,a_t) = R(s_t,a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1},a_{t+1}), \qquad (1)$$

approximated via function approximation and sampling.

Then, following the setting in Q-Transformer[14], we need to apply discretization and autoregression by regarding each action as a different dimension:

$$Q(s_{t-w:t}, a_t^{1:i-1}, a_t^i) = \begin{cases} \max\limits_{a_t^{i+1}} Q(s_{t-w:t}, a_t^{1:i}, a_t^{i+1}) & \text{if } i \in \{1,\dots,d_{\mathscr{A}}-1\} \\ R(s_t,a_t) + \gamma \max\limits_{a_{t+1}^1} Q(s_{t-w+1:t+1}, a_{t+1}^1) & \text{if } i = d_{\mathscr{A}} \end{cases} \qquad (2)$$

where $\tau = (s_1,a_1,\dots,s_T,a_T)$ is a trajectory of robotic experience of length T from an offline dataset $\mathscr{D}$. t is a given time-step, and a_t is the corresponding action in the trajectory, $a_t^{1:i}$ denote the vector of action dimensions from the first dimension a_t^1 until the i-th dimension a_t^i, i can range from 1 to the total number of action dimensions $d_{\mathscr{A}}$, w is a time window of state history.

To tackle the out-of-distribution question in offline datasets, we add a conservative penalty [32] that pushes down the Q-values $Q(s,a)$ for any action a outside of the dataset, thus ensuring that the maximum value action is in-distribution. In CQL, let π_β be the behavioral policy that induced a given dataset $\mathscr{D}$, and let $\tilde{\pi}_\beta(a|s) = \frac{1}{Z(s)} \cdot \big(1.0 - \pi_\beta(a|s)\big)$ be the evaluation policy. Our objective to train the Q-function is:

$$J = \frac{1}{2}\, \mathbb{E}_{s\sim\mathscr{D}, a\sim\pi_\beta(a|s)} \left[\Big(Q(s,a) - \mathscr{B}^* Q^k(s,a) \Big)^2 \right] + \alpha \cdot \frac{1}{2} \mathbb{E}_{s\sim\mathscr{D}, a\sim\tilde{\pi}_\beta(a|s)} \left[(Q(s,a) - 0)^2 \right], \qquad (3)$$

where the first term trains the Q-function by minimizing the temporal difference error objective as defined in Eq. 2, and the second term regularizes the Q-values to the minimal possible Q-value of 0 in expectation under the distribution of actions induced by $\tilde{\pi}_\beta$, which we denote as a conservative regularization term $\mathscr{L}_C$, α is a factor which modulates the strength of the conservative regularization.

IV. Methods

A. Auto-collected Quadruped Robot Datasets

To effectively train a generalist model through RL, it is essential to facilitate the seamless collection of a diverse dataset, including successful data and failed data, enabling corrective feedback and scalable task evaluation Therefore, we collect a large-scale multi-task dataset, **QUARD-Auto**, which includes multiple tasks such as navigation and whole-body manipulation. Next, we will discuss the main components of our data collection process.

Environment and Tasks. In this paper, we define and collect the data of 5 kinds of tasks. The detailed list of tasks in the training dataset is shown in Table I. The data was collected in Nvidia's Isaac Gym [47], a powerful simulator that allows us to collect massive robot trajectories in parallel. More statistical details about QUARD-Auto can be seen

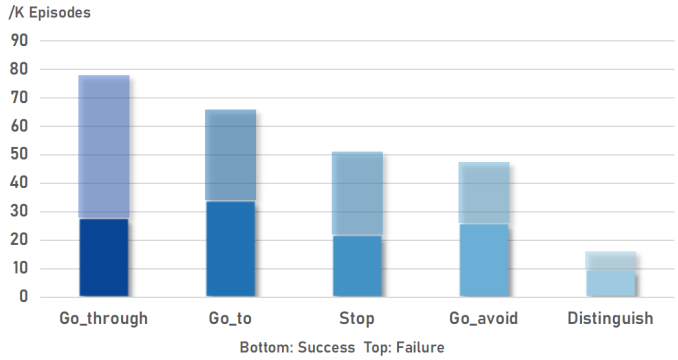

Fig. 3: **Statistic of QUARD-Auto.** The **Bottom** parts denote the successful tasks; the **Top** parts denote the failed tasks.

Parameter	Value
action_dim	12
num_layers	8
layer_size	4096
num_heads	8
num_kv_heads	8
context_len	512
time_length	7
vocab_size	256
num_experts	8
top_k_experts	2

TABLE II: **Model architecture.**

in Figure 3. Different tasks correspond to different success criteria. For example, in the "*Go to*", "*Go avoid*", and "*Go through*" tasks, the success condition is to reach a specified location. The success condition for "*Stop*" is to touch and stop the moving object and the success condition for "*Distinguish*" is to turn to the selected visual target.

Data Collection. For simulated data collection, the robot uses a combination of low-level and high-level control. The high-level control combines path planning with robot locomotion according to the global spatial information of the robot, obstacles, and target objects. For autonomous collection, we directly utilize a pre-trained policy to eliminate any need for manual teleoperation or specific trajectory design. Here, we utilize GeRM w/o MoE pre-trained on demonstrations as our high-level policy, which can receive instructions (from a simple pre-written template) and images (from a camera in the simulated environment) and output commands, eventually forming complete trajectories. The low-level control deploys the command output by the high-level policy into robot actions. Here, we adopt the approach proposed in [48] as the pre-trained low-level control strategy to output actual robot joint angles. We collected instructions, images, and command data for each frame and ultimately obtained a mix of successful and unsuccessful data.

B. Mixture-of-Experts Network

GeRM is based on a transformer architecture [49] and consists of 8 self-attention layers and 167M total parameters that outputs robot command, and the FFNs are replaced by MoE

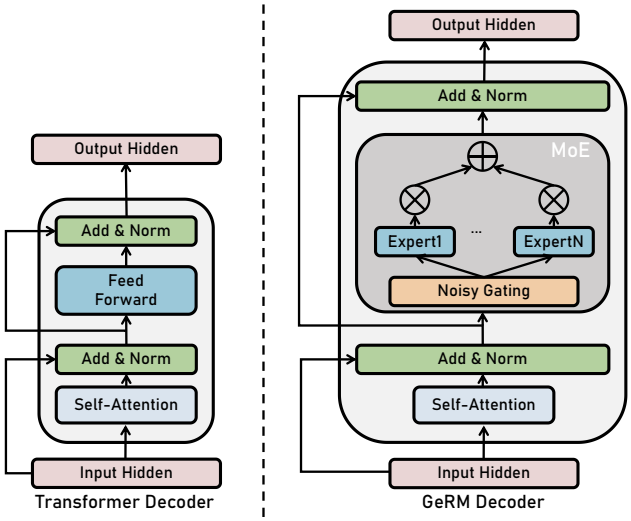

Fig. 4: **Decoder Structure. Left:** Conventional Transformer Decoder; **Right:** GeRM Transformer Decoder with MoE Module.

layers. The model architecture parameters are summarized in Table II.

We present a brief overview of the Mixture-of-Experts layer in Figure 4. The MoE module's output for a given input x is computed through the weighted sum of the expert networks' outputs, where the weights are given by the gating networks G. Then the output y could be described as:

$$y = \sum_{i=0}^{n-1} G(x)_i \cdot E_i(x), \tag{4}$$

where n is the number of expert network, the $G(x)_i$ denotes the n-dimensional output of the gating network for the i-th expert, and $E_i(x)$ is the output of the i-th expert network. There are multiple alternative ways of implementing G [50], [18], and one simple but effective way is implemented by taking the softmax over the Top-K logits of a linear layer. Before taking the softmax function, we add tunable Gaussian noise, which helps with load balancing - the Gaussian noise term adds randomness while making the process of obtaining discrete quantities from continuous quantities differentiable, thereby allowing for the back-propagation of gradients. We use

$$\begin{aligned} G(x) &= \text{Softmax}(K(H(x),k)) \\ &= \frac{\exp(k(x)_i)}{\sum_{j=0}^{N-1} \exp(k(x)_j)} \\ &\text{for} \quad i = 0,1,2,...,n-1, \end{aligned} \tag{5}$$

$H(x)$ is implemented by

$$H(x)_i = (x \cdot W_g)i + \mathcal{N}(0,1) \cdot \text{Softplus}((x \cdot W_{\text{noise}})_i), \tag{6}$$

where W_g denotes the weights of gates, and $K(x,k)$ is implemented by

$$\begin{aligned} K(x,k) &= \text{TopK}(x \cdot W_g) \\ &= \begin{cases} x \cdot W_g, & \text{if } x \text{ is in the TopK elements.} \\ -\infty, & \text{otherwise.} \end{cases} \end{aligned} \tag{7}$$

where k in TopK denotes the number of experts used per token, it is a hyperparameter that modulates the amount of compute used to process each token. When n is changed while K is fixed, the model's parameters could be changed while its computational cost is still constant. Therefore we also called the model's total parameter count the sparse parameter count and the parameters for processing an individual token the active parameter count, which means parameters actually used when inference.

C. Vision-Language-Action Model in Reinforcement Learning

An overview of GeRM is shown in Figure 1. In GeRM, the instruction is first processed via universal sentence encoder [51] $E_{text}(z_i|s)$ to get 512-dimension vectors z_i, then sent into the ImageNet-pretrained EfficientNet-B3 [52] with FiLM [53] $q_v(z_v|s,z_i)$ together with the history of 6 (the 7th image only for calculating Q-value) images w to got vision-language tokens z_v. The resulting vision-language tokens z_v are followed by a TokenLearner [54] $\tau(t|z_v)$ to compute a compact set of tokens t, and finally MoE Transformer decoders $p_{MoE}(a_d|t)$ described in IV-B to attend over these tokens and produce discretized action tokens a_d. We follow the RL method described in III to renew MoE Transformer decoders. The policy GeRM could be shown as follows:

$$\text{GeRM}(a_d|s,w) = p_{MoE}(a_d|t)\tau(t|z_v)q_v(z_v|w,z_i)E_{text}(z_i|s) \tag{8}$$

where s,w are the input images and language instruction and q_v are the language-image feature encoder, τ represents the token-learner and p_{MoE} indicates the transformer decoder to output action a_d. Eventually a_d is de-tokenized into **12**-dimensional commands:

$$\left[v_x, v_y, \omega_z, \theta_1, \theta_2, \theta_3, f, h_z, \phi, s_y, h_z^f, T\right] \tag{9}$$

Here, v_x, v_y, and ω_z represent the velocities along the x-axis, y-axis, and z-axis respectively. θ_1, θ_2, and θ_3 indicate the gait pattern, f denotes the frequency, h_z represents the height of the robot, ϕ denotes the pitch angle, s_y corresponds to the foot width, h_z^f represents the foot height, and T indicates the termination signal of the action.

V. Experiments

In our experiments, we aim to answer the following questions: **Q1.** How does the effectiveness of GeRM as a generalist model, which learns from a combination of demonstrations and sub-optimal data? **Q2.** How important are the specific designs (MoE module, Q-learning) in GeRM? **Q3.** Does the MoE module leverage its strength in size and efficiency in GeRM? **Q4.** How does GeRM demonstrate its advantages in training efficiency and data utilization? **Q5.** Can GeRM exhibit emergent skills across different tasks?

A. Experiments Setup

Offline Training Datasets. The offline dataset used in our experiment includes two categories: demonstrations and sub-optimal data. Demonstrations correspond to successful tasks, which consist of 5 types of tasks, 99 sub-tasks, with a total of **8610** trajectories and **2238600** vision-language-action sets, the average length of each trajectory is 260 frames, all sourced from human demonstration data in QUARD [21]. Sub-optimal data in our context refers to data that is auto-collected and demonstrates efforts to complete tasks, but ultimately result in failure. This part of data comprises 5 types of tasks, 99 sub-tasks, and includes a total of **2766** trajectories and **1,548,960** vision-language-action sets. Each trajectory averagely spans 560 frames, with all data sourced from QUARD-Auto.

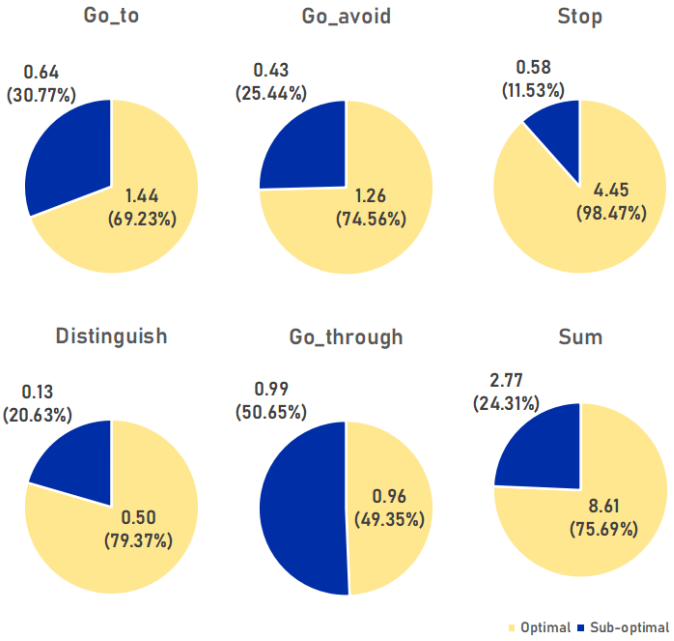

Fig. 5: **Training dataset.** The ratio of the **optimal trajectories** and **sub-optimal trajectories** used in training.The unit of trajectory number in the graph is K=10^3.

Please note that as an efficient model for data utilization, GeRM's training does not necessitate the use of all the data in the dataset. This could ensure a fair comparison between GeRM and other imitation learning methods for they shared the fully same successful data. To fully harness the learning potential of RL within sub-optimal data, we establish a ratio of demonstration to sub-optimal data at **75.69%** and **24.31%**, respectively. For simplicity, we design sparse rewards: the reward of demonstration is **1.0**, and sub-optimal data is **0.0**. More detail can be seen in Figure 5.

Low-level controller. We use the RL policy trained in walk-these-ways[48] as the low-level controller for our robot. In walk-these-ways, a controller capable of controlling various gaits and body instructions was trained using RL methods with a mix of different reward functions related to diverse gait and body state commands. The policy was trained in parallel across a large number of simulation environments and different environmental parameters. By randomizing various commands within a certain range, the robot's ability to execute a variety of commands is fully trained. By adjusting these command parameters , complex behaviors can be generalized during inference.

Baseline. To evaluate the effectiveness of GeRM and the

Model	Total Params	Active Params	Sub-optimal Data	Go_to	Go_avoid	Stop	Distinguish	Go_through
RT-1	33.50M	33.50M	N	48.67	33.50	42.5	44.33	0
GeRM w/o RL	83.48M	39.31M	N	49.37	46.37	44.88	52.00	28.44
GeRM w/o MoE	33.50M	33.50M	N	55.01	55.44	43.93	60.73	35.34
			Y	62.43	60.89	45.67	63.55	47.79
GeRM	83.48M	39.31M	N	86.37	**87.36**	50.31	75.50	73.66
			Y	**90.50**	85.50	**71.00**	**82.50**	**75.00**

TABLE III: **Multi-task performance comparison. GeRM** outperforms other models on most tasks while using approximately the same active parameters. The numbers in the table represent the success rate of tasks (%) .

necessity of the existence of MoE structure and Q-Learning. We select 2 IL approaches (RT-1[10], GeRM w/o RL) and 1 RL approach (GeRM w/o MoE) as our baseline.

Here we adjust RT-1 to suit the quadruped robots. The specific approach involves changing the model's output from the 8-dimensional commands (x, y, z, row, pitch, yaw, gripper, terminate) of a 7 DoFs robotic arm to the 12-dimensional commands (Equation 9) required for the low-level control of a quadruped robot, and appropriately increasing the number of parameters in each FFN layer. GeRM w/o RL is our GeRM trained in an imitation learning way instead of RL way and GeRM w/o MoE is GeRM ablating the MoE structure.

Evaluation Details. We conducted a comprehensive and robust series of experiments. To ensure data fidelity and mitigate the impact of stochastic variability, our primary experiments for each model encompassed the entirety of tasks including all **99** sub-tasks, with **400** trajectories meticulously tested for each. To evaluate **Q1**, we evaluate GeRM on different settings of gaits, such as "trotting", "bounding", "pronking", and "pacing", and different object settings, including seen objects that exist in offline datasets and unseen objects that out of the distribution, to test its performance as a generalist model. In the experiments pertaining to **Q4**, 400 trajectories were rigorously evaluated per epoch for each model on a single task. Additionally, a subset of experiments was allocated for other necessary activities (e.g. computational cost analysis and visualization). Furthermore, employing the autonomous data-collection methodology discussed earlier, we systematically gathered all testing data to facilitate the expansion of our dataset.

B. Experimental Results

Q1&Q2. GeRM effectively learns from mix-quality data, outperforms other methods, and demonstrates superior capabilities in multi-tasks with MoE Module and Q Learning playing significant roles in GeRM. The experimental results in Table III aim to answer Q1&Q2. Since there is only a maximum of **8610** demonstrations of different tasks, we observe from Table III that an IL algorithm like RT-1 and GeRM w/o RL, which also uses a similar Transformer architecture, struggles to obtain a good performance when learning from the limited pool of demonstrations. Offline RL method (GeRM w/o MoE), can learn from both demonstrations and sub-optimal episodes, and show better performance compared to RT-1. Indeed, GeRM trained on demonstrations has exhibited a significant performance improvement, thanks to the model architecture of GeRM itself. Furthermore, GeRM trained with the inclusion of sub-optimal data has further enhanced its performance across most tasks, particularly achieving substantial improvements in "*Stop*" tasks. GeRM has the highest success rates and outperforms both the behavior cloning baseline (RT-1, GeRM w/o RL) and offline RL baselines (GeRM w/o MoE), exceeding the performance of the best-performing prior method by **30%-70%**. This demonstrates that GeRM can effectively improve upon human demonstrations using autonomously collected sub-optimal data. It also demonstrates the significance of each component design within GeRM.

Q3. MoE Modules balance computational cost and performance by activating part of the parameter when inference. We also compare the parameter counts of each model. GeRM exhibits efficiency in the cost-performance spectrum (see Table III). As sparse Mixture-of-Experts models, GeRM w/o RL and GeRM only use **39.31M** activated parameters for each token, which means it only uses **1/2** total parameters and **1/8** FFN layers. With slight parameter increases (only **5.81M**), GeRM is able to outperform RT-1 across all categories. Moreover, another MoE model GeRM w/o RL performs better than RT-1 across most categories with the same activated parameters.

Note that this analysis focuses on the active parameter count, which is directly proportional to the inference computational cost, but does not consider the hardware utilization and training costs. As for device utilization, we note that the MoE layer introduces additional overhead due to the routing mechanism and the increased memory loads when running more than one expert per device. They are more suitable for batched workloads where one can reach a good degree of arithmetic intensity. For training cost, we will discuss it in the next question.

Q4. GeRM exhibits commendable training efficiency. While GeRM could control its computational cost at a relatively rational level, its efficiency in the training stage may raise concerns. So we perform a comparison experiment between GeRM and other baselines to assess their performance in the "*Go to the red cube*" task. To ensure the same

input data volume, we only utilize the demonstration data to exclude potential additional data volume (sub-optimal data). According to Figure 6, under the same number of epochs, GeRM often achieves higher success rates. By the **2**nd epoch, it has already reached a similar level to that of RT-1's **20**th epoch and essentially converged by the **7**th epoch. Similarly, GeRM w/o MoE, also an offline RL method, converges in approximately **8** epochs. In contrast, Imitation Learning Methods (GeRM w/o RL, RT-1) fail to converge by the **10**th epoch. It is noteworthy that GeRM's performance, even when exclusively trained with demonstrations, remains impressive. This observation underscores GeRM's proficiency not only in effectively harnessing sub-optimal data but also in leveraging demonstrations with superior efficiency compared to alternative methodologies. **Such findings serve to further substantiate the efficacy of GeRM in optimizing data utilization strategies.**

Q5. GeRM shows emergent skills in dynamic adaptive path planning. Through the RL from the large-scale combination of demonstrations and sub-optimal data, GeRM has the potential to autonomously explore unseen skills beyond the demonstrations, known as emergent skills. Therefore, we aim to evaluate the degree to which such models can show emergent skills. We demonstrate an example in Figure 2. Taking the task "*Go to the fan and avoid the obstacle*" as an example, in the upper figure, the quadruped robot's vision is limited at the initial position, hampering its ability to determine the direction of movement. To avoid the obstacle it turns to the left randomly. Subsequently, upon encountering the incorrect visual input, the robot executes a substantial reorientation to align with the correct target outside its original field of view. It then proceeds to steer towards the destination, ultimately accomplishing the task. Notably, such trajectories were out-of-distribution of our training dataset. Conversely, the lower figure illustrates a common failure example by IL ways, the robot chooses the false direction and directly reaches the wrong target. We find that through our exploration GeRM inherits novel capabilities in terms of **dynamic adaptive path planning** in the context of the scene, which means it can make decisions, plan future paths, and change next-step action according to the visual perception.

VI. Conclusion, Limitations and Future Work

We have presented GeRM, the first Mixture-of-Experts model for quadruped reinforcement learning. We have surpassed the limitations of quadruped robots in demonstration by using RL, enhancing the ability and efficiency of data utilization, with the potential to elevate robot performance to super-human levels. By incorporating the transformer-based MoE model, we have expanded the model's capacity and reinforced its capabilities, enabling it to possess generalist abilities in multi-task. Our model achieves high performance with the limited computational cost, while further improving the data utilization capabilities and fostering the development of emergent skills. We introduce QUARD-Auto, a dataset comprising both successful and failed task data, totaling 257k trajectories, serving as a benchmark for robotic imitation learning and reinforcement learning in the future, which could benefit the robot learning community.

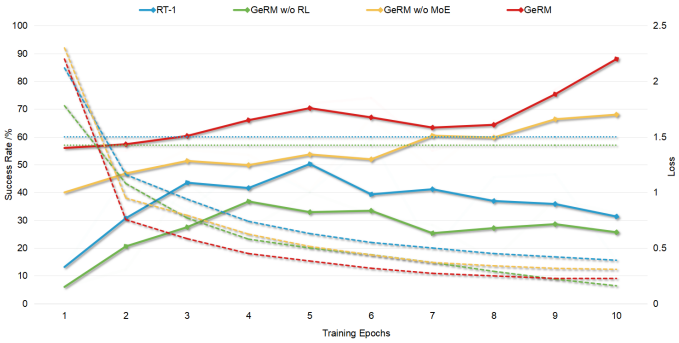

Fig. 6: **Performance change and Loss on "*Go to the red cube*" task.** Solid lines represent the success rate, dotted lines represent the final success rate for 20 epochs, and dashed lines represent loss. Note: RL approaches employ MSE loss, which should be scaled by 0.1, while IL ways employ Cross-Entropy as the loss function.

Limitations & Future Work. 1. While our model demonstrates effectiveness for quadruped robots in simulation, our next step involves extending its capabilities to real-world scenarios. We aim to assess its performance in real-world environments and conduct additional research to ensure its adaptability to real-world settings. **2.** With aspirations for our model, GeRM, to excel across a broader range of tasks as a generalist, our future endeavors involve expanding its proficiency. To achieve this, we intend to curate a larger dataset encompassing a wider array of task categories. This will enable us to further evaluate the robustness of GeRM and its ability to generalize effectively.

VII. Acknowledgement

This work was supported by NSFC General Program (Grant No. 62176215), the National Science and Technology Innovation 2030 - Major Project (Grant No. 2022ZD0208800), and the National Key R&D Program of China(No. 2704700).

References

[1] Hutter *et al.*, "Anymal - a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 38–44.

[2] S. Lyu, H. Zhao, and D. Wang, "A composite control strategy for quadruped robot by integrating reinforcement learning and model-based control," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 751–758.

[3] Karnan *et al.*, "Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 807–11 814, 2022.

[4] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science Robotics*, vol. 5, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:224828219

[5] S. Choi, G. Ji, J. Park, H. Kim, J. Mun, J. H. Lee, and J. Hwangbo, "Learning quadrupedal locomotion on deformable terrain," *Science Robotics*, vol. 8, no. 74, p. eade2256, 2023. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.ade2256

[6] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang, "Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers," *ArXiv*, vol. abs/2107.03996, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235765481

[7] S. Jeon, M. Jung, S. Choi, B. Kim, and J. Hwangbo, "Learning whole-body manipulation for quadrupedal robot," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 699–706, 2024.

[8] D. Kalashnikov, J. Varley, Y. Chebotar, B. Swanson, R. Jonschkowski, C. Finn, S. Levine, and K. Hausman, "Mt-opt: Continuous multi-task robotic reinforcement learning at scale," *arXiv: Robotics,arXiv: Robotics*, Apr 2021.

[9] A. Kumar, A. Singh, F. Ebert, Y. Yang, C. Finn, and S. Levine, "Pre-training for robots: Offline rl enables learning new tasks from a handful of trials," Oct 2022.

[10] A. Brohan *et al.*, "Rt-1: Robotics transformer for real-world control at scale," 2023.

[11] O. X.-E. Collaboration *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:266359827

[12] F. Ebert *et al.*, "Bridge data: Boosting generalization of robotic skills with cross-domain datasets," *arXiv preprint arXiv:2109.13396*, 2021.

[13] H. Walke *et al.*, "Bridgedata v2: A dataset for robot learning at scale," *arXiv preprint arXiv:2308.12952*, 2023.

[14] Chebotar *et al.*, "Q-transformer: Scalable offline reinforcement learning via autoregressive q-functions."

[15] Brown *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[16] A. Brohan *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," 2023.

[17] S. Wang, M. Han, Z. Jiao, Z. Zhang, Y. N. Wu, S.-C. Zhu, and H. Liu, "Llm3:large language model-based task and motion planning with motion failure reasoning," 2024. [Online]. Available: https://arxiv.org/abs/2403.11552

[18] 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.

[19] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton, "Adaptive Mixtures of Local Experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 03 1991. [Online]. Available: https://doi.org/10.1162/neco.1991.3.1.79

[20] M. I. Jordan and R. A. Jacobs, "Hierarchical Mixtures of Experts and the EM Algorithm," *Neural Computation*, vol. 6, no. 2, pp. 181–214, 03 1994. [Online]. Available: https://doi.org/10.1162/neco.1994.6.2.181

[21] P. Ding, H. Zhao, Z. Wang, Z. Wei, S. Lyu, and D. Wang, "Quar-vla: Vision-language-action model for quadruped robots," *ArXiv*, vol. abs/2312.14457, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:266520894

[22] N. Jaques, A. Ghandeharioun, J. Shen, C. Ferguson, A. Lapedriza, N. Jones, S. Gu, and R. Picard, "Way off-policy batch deep reinforcement learning of implicit human preferences in dialog." *arXiv: Learning,arXiv: Learning*, Jun 2019.

[23] Y. Wu, G. Tucker, and O. Nachum, "Behavior regularized offline reinforcement learning," *arXiv: Learning,arXiv: Learning*, Sep 2019.

[24] X. Peng, A. Kumar, G. Zhang, and S. Levine, "Advantage-weighted regression: Simple and scalable off-policy reinforcement learning," *Cornell University - arXiv,Cornell University - arXiv*, May 2021.

[25] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," *arXiv: Learning,arXiv: Learning*, Oct 2021.

[26] S. Fujimoto and S. Gu, "A minimalist approach to offline reinforcement learning," *Neural Information Processing Systems,Neural Information Processing Systems*, Dec 2021.

[27] X. Chen, Z. Zhou, Z. Wang, W. Che, Y. Wu, and K. Ross, "Bail: Best-action imitation learning for batch deep reinforcement learning," *arXiv: Learning,arXiv: Learning*, Oct 2019.

[28] H. Furuta, Y. Matsuo, and S. Gu, "Generalized decision transformer for offline hindsight information matching."

[29] L. Meng, M. Wen, C. Le, X. Li, D. Xing, W. Zhang, Y. Wen, H. Zhang, J. Wang, Y. Yang, and B. Xu, "Offline pre-trained multi-agent decision transformer."

[30] L. Liu, Z. Tang, L. Li, and D. Luo, "Robust imitation learning from corrupted demonstrations."

[31] Y. Zheng, L. Li, F. Asano, C. Yan, X. Zhao, and H. Chen, "Modeling and analysis of tensegrity robot for passive dynamic walking," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 2479–2484.

[32] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," *arXiv: Learning,arXiv: Learning*, Jun 2020.

[33] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *arXiv: Learning,arXiv: Learning*, Jan 2017.

[34] Hestness *et al.*, "Deep learning scaling is predictable, empirically," *arXiv: Learning,arXiv: Learning*, Dec 2017.

[35] 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," *Cornell University - arXiv,Cornell University - arXiv*, Jun 2020.

[36] B. Zoph, "Designing effective sparse expert models," in *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2022, pp. 1044–1044.

[37] N. Du *et al.*, "Glam: Efficient scaling of language models with mixture-of-experts," in *International Conference on Machine Learning*. PMLR, 2022, pp. 5547–5569.

[38] A. 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.

[39] D. Dai, C. Deng, C. Zhao, R. Xu, H. Gao, D. Chen, J. Li, W. Zeng, X. Yu, Y. Wu, *et al.*, "Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models," *arXiv preprint arXiv:2401.06066*, 2024.

[40] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," *ArXiv*, vol. abs/2109.12098, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:237396838

[41] S. Reed, Zolna, *et al.*, "A generalist agent," *arXiv preprint arXiv:2205.06175*, 2022.

[42] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," in *Conference on Robot Learning*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247618840

[43] P. Li, T. Liu, Y. Li, M. Han, H. Geng, S. Wang, Y. Zhu, S.-C. Zhu, and S. Huang, "Ag2manip: Learning novel manipulation skills with agent-agnostic visual and action representations," 2024. [Online]. Available: https://arxiv.org/abs/2404.17521

[44] X. Li *et al.*, "Vision-language foundation models as effective robot imitators," *ArXiv*, vol. abs/2311.01378, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:264935429

[45] A. Szot, M. Schwarzer, H. Agrawal, B. Mazoure, W. Talbott, K. Metcalf, N. Mackraz, D. Hjelm, and A. Toshev, "Large language models as generalizable policies for embodied tasks," *arXiv preprint arXiv:2310.17722*, 2023.

[46] H. Zhao, M. Zhang, W. Zhao, P. Ding, S. Huang, and D. Wang, "Cobra: Extending mamba to multi-modal large language model for efficient inference," *ArXiv*, vol. abs/2403.14520, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:268553791

[47] V. Makoviychuk *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," 2021.

[48] G. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," in *Conference on Robot Learning*. PMLR, 2023, pp. 22–31.

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[50] Y. Zhou, T. Lei, H. Liu, N. Du, Y. Huang, V. Zhao, A. M. Dai, Q. Le, J. Laudon, *et al.*, "Mixture-of-experts with expert choice routing," *Advances in Neural Information Processing Systems*, vol. 35, pp. 7103–7114, 2022.

[51] D. Cer, Yang, *et al.*, "Universal sentence encoder," *arXiv: Computation and Language,arXiv: Computation and Language*, Mar 2018.

[52] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from rgbd images: Learning using a new rectangle representation," in *2011 IEEE International conference on robotics and automation*. IEEE, 2011, pp. 3304–3311.

[53] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. C. Courville, "Film: Visual reasoning with a general conditioning layer," in *AAAI Conference on Artificial Intelligence*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:19119291

[54] M. Ryoo, A. Piergiovanni, A. Arnab, M. Dehghani, and A. Angelova, "Tokenlearner: Adaptive space-time tokenization for videos," *Neural Information Processing Systems,Neural Information Processing Systems*, Dec 2021.