Learning Natural Locomotion from Demonstration

Lu Liu and Michael Drolet

Abstract— **Learning-based controllers have reached remarkable results on different platforms due to the recent improvements seen in robotics and artificial intelligence. As a result, quadrupeds have gained much attention due to their ability to adapt to challenging terrains (compared to other types of robots). Mastering agile, robust, and natural locomotion in real-world environments presents significant challenges due to the complexities of non-simulated environmental dynamics. Recent studies demonstrate that conventional reinforcement learning (RL) methods can successfully learn difficult tasks. Nevertheless, the resulting behaviors tend to be energy-inefficient and artificial due to the overreliance on task rewards. This paper centers on examining the concept of style reward, which has emerged in recent years as a crucial element in enabling RL-based controllers to exhibit natural locomotion.**

I. INTRODUCTION

Compared to wheeled robots, legged robots possess superior capabilities in adapting to challenging terrains, such as on stairs and discontinuous ground. This adaptability is achieved through the strategic planning of contact points. However, generating natural, agile, and robust behavior in tasks remains a significant challenge for legged robots. Traditional methods, which involve manually designing control strategies, demand extensive knowledge of the dynamic systems involved and considerable skill. This design process is not only time-consuming but also results in control strategies that lack adaptability when applied to new tasks, requiring a redesign of the controller for each new application. To address the limitations of these manually crafted controllers, reinforcement learning (RL) offers an efficient solution by enabling an agent to learn and update control strategies through environmental interaction. Despite this, the motions produced via RL are often physically infeasible, energy-inefficient, and exhibit unnatural behavior during task execution. To address these challenges, many researchers have begun incorporating an additional reward into the reward function. This supplementary reward is designed to motivate the agent towards desired behaviors, thereby guiding the reinforcement learning (RL) algorithm to encourage the agent to emulate reference motions while achieving task objectives.

The essence of natural behavior learning lies in formulating reward functions, which consist of style rewards and task rewards. The style objective defines a motion imitation reward, which can be hand-crafted through motion tracking or learned automatically through adversarial imitation learning. This component ensures that the reproduced behavior is similar to the reference motions. The objective is introduced to guarantee that the overarching goal or task is fulfilled. By combining these two rewards, the reward function can direct the learning process to generate such a control strategy, reproducing natural, agile, and robust behaviors.

One prominent RL algorithm employed for training in this framework is Proximal Policy Optimization (PPO) [16]. PPO is derived from Trust Region Policy Optimization (TRPO), utilized in Generative Adversarial Imitation Learning (GAIL) [8]. In policy iterations, we need to compute policy gradients, which help us better evaluate the executed actions under the current policy. However, without constraints on the magnitude of policy updates during optimization, it often results in excessively large adjustments. This divergence can adversely affect the learning progression and the overall effectiveness of the policy improvement. Instead of limiting the Kullback–Leibler (KL) divergence between the old and new policies, PPO clips the probability ratio to achieve a restriction similar to TRPO, therefore having the stability and reliability of trust-region methods but are much simpler to implement.

This report primarily discusses various methods for extracting style rewards to facilitate the generation of natural motions in four-legged robots for specific tasks. An overview of the report is depicted in Figure 1. Initially, we talk about the dominant platforms for quadrupedal robots and outline the main research objectives. To understand the current learning-based control strategies for quadrupedal robots, we discuss recent works with reinforcement learning (RL) and the methods for transferring simulation results to real-world applications. The report focuses on extracting style rewards, either through motion tracking or adversarial motion priors, and then integrating these style rewards into the task rewards. By employing reinforcement learning, we aim to develop a policy capable of completing tasks with a specific style. In the end, we discuss the future work and possible limitations of style reward.

II. QUADRUPED LOCOMOTION

One of our primary interests in this seminar is to explore how to learn agile and realistic locomotion on robots. As this comprises our key interest, there are various platforms of interest, such as bipedal and quadruped robots. While there are many opportunities for improving bipedal locomotion, this paper focuses on the frontier of developments in quadruped research.

A. Platforms

A variety of quadrupedal robots have stood on the stage in the past decades. ANYmal from ETH is one of the most promising four-legged systems [24], which features outstanding mobility and high flexibility. ANYmal has a

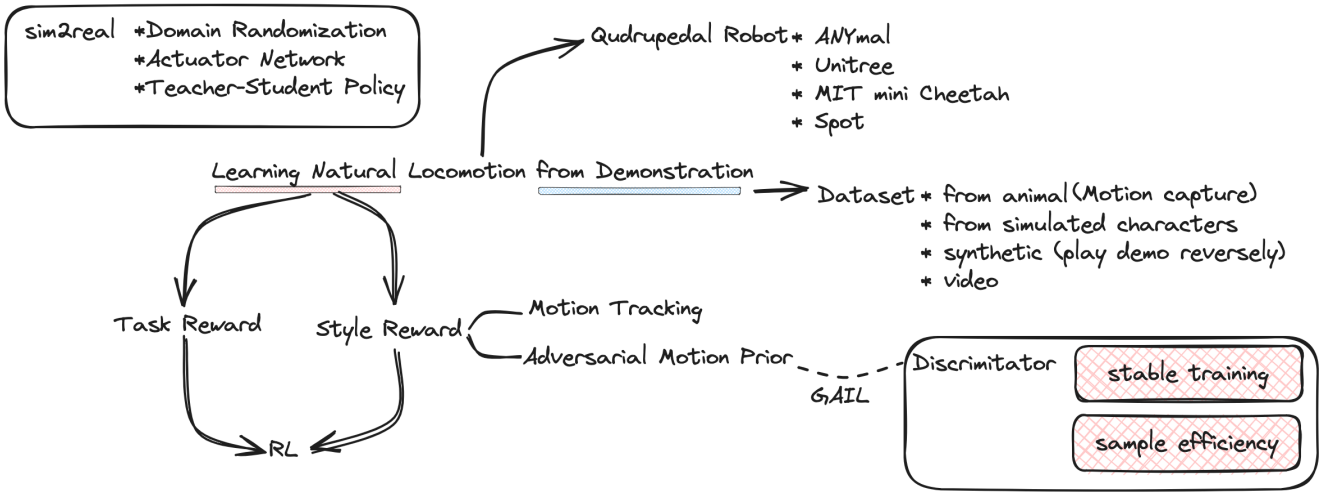

Fig. 1. An Overview of "Learning Natural Locomotion from Demonstration"

different topology design on the leg compared to others. The full rotation of joints allows it to achieve various configurations for different tasks like walking and stair climbing. The recent research based on it exhibits more powerful performance on robustness over challenging terrain [14]. The most prominent platform seems to be Unitree, a series of quadrupedal robots from Unitree Robotics [25] that is quite popular among academic laboratories. As showcased in the parkour demos [19], over challenging terrain [23], in the wild [15] and in other versatile motion task [3], [20]–[22], [28]–[31]. MIT mini cheetah is a lightweight and mechanically robust quadrupedal robot, which is also the first one to finish motion backflip [26]. Additionally, Boston Dynamics' Spot has garnered a lot of attention due to its exceptional skills in various scenarios, including dancing and dragging a truck. However, the development of its low-level controller is challenging because of proprietary restrictions. Spot is mainly a commercial product that has been deployed in rescue operations, searches, and examinations in dangerous places [27].

B. Research Objectives

The research on quadrupeds focuses on the practical applications of the robot in real-life scenarios. In real life, dog-like robots are often required to perform various tasks in common scenarios, such as slopes with different inclinations, irregular gravel fields, and smooth surfaces with varying levels of slipperiness, like ground covered by mud, snow, or ice. As showcase in [14] [23]. They may even encounter discontinuous ground, like steps and gaps, which usually require them to climb and jump [19], [28]. While these may seem trivial for humans and animals, they pose significant challenges for robots. Therefore, studying how quadrupedal robots maintain stability over different terrains is a crucial research direction.

Meanwhile, the tasks often require robots to exhibit a variety of gaits like walking, trotting, and galloping, as showcased in [31]. Robots need to adopt different gaits to achieve varying speeds. Investigating how to make dogs perform different gaits for given tasks and stabilize transitions between different gaits is also an essential research direction [28]. In industrial applications, we not only require dogs to complete tasks safely and stably but also wish for them to do so effortlessly, i.e., low-power transport, which is a characteristic of natural locomotion. as showcased in [3], [15], [20].

III. RELATED WORK

A. Learning Natural Locomotion from demonstration

Learning from demonstration, also known as imitation learning, offers distinct advantages over reinforcement learning by leveraging expert demonstrations. It eliminates the need to explicitly design a reward function, which is usually tedious, time-consuming, and challenging, especially under real-world scenarios. Imitation learning incorporates expert experience by aligning the agent's policy with the expert, aiming to replicate expert behaviors. Over the past decades, various approaches have emerged, broadly classified into three categories: behavior cloning, inverse reinforcement learning, and adversarial inverse reinforcement learning. Behavior cloning achieves this alignment through supervised learning. However, it tends to perform poorly in unseen scenarios due to generalization issues. Inverse reinforcement learning (IRL) aims to infer the underlying reward function based on the observed behavior of the expert. The reconstructed reward function guides the learning process, hence the agent is enabled to adopt an expert-like policy. Adversarial inverse reinforcement learning (AIRL) builds on IRL by integrating adversarial training, where the policy and a discriminator are trained simultaneously. The agent strives to mimic expert behaviors to deceive the discriminator, while the discriminator seeks to differentiate between the actions of the agent and those of the expert. This adversarial

process encourages the agent to generate behaviors that are indistinguishable from the expert.

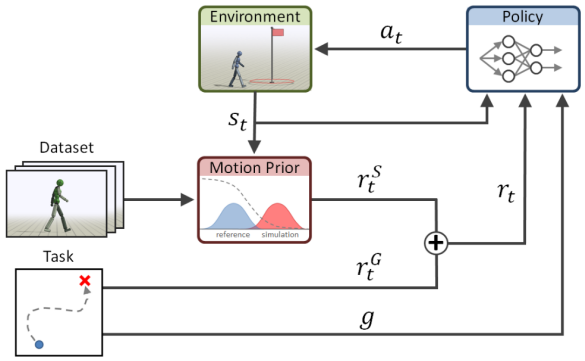

Fig. 2. Schematic overview of Learning natural locomotion with a style reward. Starting with motion reference that outlines desired motion styles for the agent, the system then trains a motion prior, specifying style reward r_t^S. The style reward is then combined with the task reward, denoted as r_t^G, and used to train a policy. By doing so, the agent can achieve task-specific goals while also emulating behaviors that are similar to those observed in the reference motion.

In the application of quadrupedal robots, employing an RL-based controller can effectively complete several tasks. However, it is common to observe that the behaviors demonstrated in given tasks often exploit unnatural gaits, result in excessive contact force, and incur high energy consumption.

To address these problems, several works have focused on integrating natural motion styles into tasks. One of the primary directions is extracting style from demonstrations. In the DeepMimic [6], the authors propose a method that combines a motion-imitation objective with task objectives, enabling animated characters to perform highly dynamic actions such as flips and spins. The motion imitation objective acts as a style reward, constructed based on the difference between the agent's joint states and the reference motion clips. It guides the learning process toward developing a policy that not only fulfills the task but also replicates the style from the motion clips. In [13], the authors provide a framework that allows robots to learn skills from real animals. The challenge in learning from motion capture data lies in the reduction from the real animal to the robot, where the real animal usually has a different topology and more flexible configuration. They address this through motion retargeting, which involves tracking key points of a real dog, such as the feet and hips, and then leveraging inverse kinematics to compute a configuration for the joints of the robot. The resulting policy enables the Unitree robot to exhibit a diverse set of skilled locomotions, including trotting, pacing, agile turning, and spinning. The aforementioned works utilize motion tracking to generate a broad spectrum of skills. However, when dealing with large, diverse, and unstructured datasets, these approaches necessitate extra effort to select appropriate motions. This challenge led to the development of the Adversarial Motion Prior (AMP) [4], which adopts a GAN-style approach to extract style rewards without relying on handcrafting. The discriminator is trained simultaneously with the RL controller, and it also provides auxiliary information in the task reward, known as the style reward. This enables animated characters to not only complete the task but also to exhibit the styles from reference motion clips. The overview of this kind of approach is shown in figure 2. For varying styles, AMP introduces a selector mechanism, allowing users to interact with the agent to produce a desired style for achieving the task goal, or to automatically select the best style based on the maximum reward achieved. The subsequent work [2] extends this approach to the wheeled-legged robot, ANYmal, discussing the adjustment of multiple adversarial motion priors. By employing a one-hot encoding selector, it achieves intentional style switching, allowing for the selection of motion priors to a given scenario. To enhance the existing dataset, they mirror the reference motion, which later proved to be a crucial factor for enabling their resulting policy on ANYmal to transition from a quadrupedal configuration to a humanoid form.

In [15], the authors integrate the teacher-student framework with AMP on the Unitree A1 robot and successfully enable it to exhibit a diverse range of natural gait patterns in various outdoor environments. They also enhanced the dataset by mirroring specific joint positions. Which turns out to be an important factor to enable Unitree A1 to perform gallop gait in the wild.

The answers to achieving natural motion in tasks are more than just AMP. The author in [32] provides an alternative approach by learning a kinematic generative model of human motion from example motion data, using an autoregressive conditional variational autoencoder (Motion VAE). The prediction of the pose at the next timestep depends on several stochastic latent variables. These latent variables are treated as the action space for reinforcement learning. With a goal-specific reward function, the animated human character can generate natural, goal-directed motion in tasks. Besides the work in [20] discusses another method, which also achieves natural gaits but without reference motion. The authors incorporate the energy information defined by a product of joint torque and velocities in the total task reward. It points out that energy minimization leads to the emergence of natural gaits.

B. Sim-to-Real

After obtaining a trained controller from the simulator, it's crucial to transition the learned behaviors to the real-world environment. This presents a significant challenge due to the substantial reality gap that cannot be ignored. To bridge this reality gap, we can construct more accurate simulation environments. One way to achieve this is by randomizing the physical parameters of the simulator, such as friction, mass, etc. A wide range of randomization could cover the physical configuration of reality. Training in these environments makes the trained controller more robust and helps the controller transfer more effectively to real-world scenarios. This technique is referred to as domain randomization (DR). It has become a common technique to facilitate the

TABLE I

SUMMARY OF EXISTING RESEARCH ON NATURAL LOCOMOTION LEARNING

Ref	Datasets	Learning Type	Sim2Real	Platform	Terrains
[13]	Mocap	Motion Tracking	DR & DA	Unitree Laikago	Indoor
[2]	Synthetic	AMP	DR & Acutator Net	ANYmal	Indoor
[3]	Mocap	AMP	DR	Unitree A1	Indoor
[15]	Mocap & Synthetic	AMP	DR & DA	Unitree A1	In/Outdoor
[20]	N/A	Engergy-based	DR & DA	Unitree A1	In/Outdoor
[32]	Mocap & Synthetic	Motion VAE	N/A	Animated Characters	N/A
[4]	Mocap & Synthetic	AMP	N/A	Animated Characters	N/A
[6]	Mocap & Synthetic	Motion Tracking	N/A	Animated Characters	N/A

transfer of results into reality. as showcased in [3], [13], [15]. Another approach is modeling an accurate physics system, including the actuators, as shown in [12]. This involves training an actuator network that outputs an estimated torque at joints, given a history of position errors and velocities. Essentially, it uses supervised learning to obtain an action-to-torque relationship that encompasses all software and hardware dynamics within one control loop. In addition to focusing on physics simulation configuration, the authors in [14] proposed a teacher-student policy. This leverages proprioceptive signals such as base velocity, orientation, and joint states to recover feature information containing privileged information like contact states, terrain profiles, friction coefficients, etc. In the teacher-student policy, as the name suggests, there are two policies. The teacher policy is trained using some reinforcement learning algorithm in simulation and has access to privileged information not available in the real world. Then, a proprioceptive student policy learns by imitating the teacher, having access only to proprioceptive signals, instead of privileged information. This framework has proven so effective that it even enables the robot ANYmal to demonstrate zero-shot generalization from simulation to natural environments. Subsequent works have also achieved remarkable results by utilizing this framework, as showcased in [15], [23]. The table I presents a summary of the approaches above. This table provides an overview, comparing key aspects and outcomes of each method to facilitate understanding of their differences and applications.

IV. STYLE REWARD

In this report, we focus on incorporating style features from reference motions into task rewards to achieve natural locomotion. We first discuss the categories of datasets. These can include motion capture files, which document key points on real animals such as dogs and horses, as well as synthetic data, which might be collected from existing RL controllers, optimized trajectory, and video. Synthetic data also might involve playing motions in reverse or mirroring existing datasets. Synthetic data enables the learning of unexpected behaviors, as demonstrated in [2] and [15]. Motion capture datasets are often collected from real animals with similar structures but varying limb lengths. Therefore, mapping the reference dataset to the dog model is necessary for obtaining learnable simulation datasets. This entire process is known as motion retargeting [33]. Initially, a set of source and target key points from animals will be recorded and mapped onto the robot's body structure according to their timestamps. This is achieved by constructing an optimization formulation that leverages inverse kinematics and additional constraints to determine the corresponding joint configuration.

To extract style information from datasets, this report discusses two approaches. The first is motion tracking, which involves tracking joints and other crucial points that represent the motion style. The second approach is AMP (Adversarial Motion Priors), which integrates adversarial imitation learning with reinforcement learning.

A. Motion Tracking

In the studies referenced in [6] and [13], the reward function encourages the policy to track a sequence of target pose $\hat{q}_0, \hat{q}_1, ..., \hat{q}_T$ at every timestep. The overall task objective at each timestep is defined as follows:

$$r_t = w^I r_t^I + w^G r_t^G \tag{1}$$

where r_t^I and r_t^G denote the imitation and task-goal objectives respectively, w^I and w^G represent their corresponding weights, typically determined manually. The imitation objective r_t^I is derived from the tracking loss associated with positions, velocities, and orientations of key points, such as joints and the base trunk, encouraging the agent to adhere to the specified motion sequence. Here is an example of imitation reward defined by motion tracking:

$$r_t^I = w^p r_t^p + w^v r_t^v + w^e r_t^e \tag{2}$$

where r^p is the pose reward, as defined in the following equation, minimizing the difference between the joint rotations specified by the reference motion and those of the robot at each timestep. $\hat{q}_t^j$ represents the t time local rotation of joint j from the reference motion, and q_j^t represents the robot's joint angle at time t. $\hat{q}_t^j \ominus q_t^j$ denotes the quaternion

difference. $||q||$ computes the rotation in radians.

$$r_t^p = \exp[-2(\sum_j ||\hat{q}_t^j \ominus q_t^j)||^2)] \tag{3}$$

The r_t^v and r_t^e represent the velocity reward and end-effector reward, defined similarly to the above equation by terms $\hat{\dot{q}}_t^j$,$\dot{q}_t^j$, $\hat{x}_t^e$ and x_t^e, which denote the angular velocity of joint j from the reference motion and robot , the relative position of end-effector from reference and robot respectively. Furthermore, in this work, the root pose and velocity are also considered to construct the root motion accurately.

The authors implemented the method on the Unitree Laikago, enabling it to perform a wide range of agile behaviors, including various locomotion gaits, dynamic hops, and turns. Nevertheless, this approach requires manually defining the reward function and selecting suitable motions for the agent to track in specific scenarios. This becomes challenging and impractical when dealing with large and complex unstructured datasets.

B. Adversarial Motion Prior

To address the challenges of large and complex datasets, one solution is to use a model capable of automatically extracting information from the reference motion. Instead of manually crafting imitation objectives, Peng, in his work [4], introduces fully automated approaches based on adversarial imitation learning. This method involves two models: a generator and a discriminator. The generator's job is to create motions, while the discriminator attempts to differentiate between the generated motion and expert motion. The discriminator assesses the disparities between the generated states and that of the experts. The learned discriminator serves as the imitation objective, effectively acting as a style reward. The style reward, when combined with the task reward, creates a comprehensive reward system. Consequently, after applying a reinforcement learning algorithm, the learned policy not only achieves the goal, such as an animated human kicking a stationary ball but also exhibits a specific style akin to the reference. A more intuitive overview is illustrated in 2. The whole reward function is defined as the following equation:

$$r(s_t, a_t, s_t, s_{t+1}, g) = w^G r^G(s_t, a_t, s_t, g) + w^S r^S(s_t, s_{t+1}) \tag{4}$$

where a task-specific reward is denoted as $r^G(s_t, a_t, s_t, g)$. It encourages the agent to move to the target. And $r^s(s_t, s_{t+1})$ is the style reward, which determines the agent how to reach the target. Compared to the task reward function, the style reward focuses solely on the state transition as its input. The complete objective for training the discriminator is usually defined as follows:

$$\arg\min_D \quad \mathbb{E}_{d^{\mathcal{M}}(s,s')}[(D(\Phi(s), \Phi(s')) - 1)^2]$$

$$+ \mathbb{E}_{d^{\pi}(s,s')}[(D(\Phi(s), \Phi(s')) + 1)^2]$$

$$+ \frac{w^{gp}}{2} \mathbb{E}_{d^{\mathcal{M}}(s,s')}[\|\nabla_\phi D(\phi)|_{\phi=(\Phi(s),\Phi(s'))}\|^2] \tag{5}$$

The training process often faces unstable dynamics, such as gradient vanishing and function approximation errors. Gradient vanishing occurs when the objective is defined using a sigmoid cross-entropy function like the loss defined in the original GAIL due to the sigmoid function saturation. Employing a least squares GAN, known for more stable training and higher quality outcomes, is a viable option. Nonetheless, using only least squares is not a panacea. Instability can still happen during training due to function approximation errors in the discriminator. Namely, the discriminator might assign non-zero gradients on the real data sample manifold, which causes the generator to overshoot and deviate from the data manifold. This can result in oscillations and instability during training. To address this, applying a gradient penalty on non-zero gradients on dataset samples can be effective. The aforementioned factors explain why the objective is defined as above. Through solving a least-squares regression problem, the objective aims to predict a score of 1 for samples from the dataset $\mathcal{M}$ and -1 for samples generated by the policy π. $\Phi(s)$ denotes an observation map that extracts a set of features relevant to state transitions. The objective has proven effective in many subsequent studies. As showcased in [15], [3], [2], [4]. It is also worth noting that most of these works employ off-policy training on the discriminator through a replay buffer to achieve sample efficiency. As mentioned in [7] [18]. Therefore the state transitions are collected from a replay buffer, which stores past experience from the interaction of the agent with the environment. The style reward function for training the policy can be defined by:

$$r(s_t, s_{t+1}) = \max\ [0, 1 - 0.25(D(\Phi(s), \Phi(s_{t+1})) - 1)^2] \tag{6}$$

When the discriminator classifies a state transition as positive, indicating that the generated motion resembles that of an expert. otherwise, it yields zero. To evaluate whether the policy trained with adversarial motion priors generates natural locomotion, we use the concept of cost of transport, a common metric for estimating the energy efficiency of locomotion. It is typically defined as follows:

$$\text{COT} = \sum_{\text{motors}} [\tau\dot{\theta}]^+ \tag{7}$$

where τ is the joint torque and $\dot{\theta}$ is the motor velocity. The task can involve various velocity commands. To meet these velocity demands, the robot adapts to different gaits. For instance, at lower velocities, the walking gait is typically the most energy-efficient, as reported in [20]. This serves as a criterion to evaluate whether the resulting policy can produce natural locomotion that demonstrates lower energy consumption.

V. CONCLUSION

In this report, we primarily explore two distinct approaches to extract style information from demonstrations, which then

work as part of the reward function in the reinforcement learning algorithm to develop a viable policy. The first method involves defining a loss function that tracks a sequence of key points, which subsequently serves as the reward function. The second approach employs adversarial imitation learning, simultaneously training a discriminator alongside the policy. This latter method automatically generates the reward function, eliminating the need for manually setting constraints. However, the training process in the latter approach may encounter instability and require a large number of samples from the interactions of the agent with the environment

A. Limitation

Generally, extracting style information from a dataset and using it as a reward results in good performance on different platforms. However, a small dataset might not be sufficient to support the policy in producing a natural gait for a given scenario. With a large dataset- one that encompasses multiple skills/scenarios- the challenge lies in finding the most suitable skill within the dataset for a given task. Additionally, finding the right coefficient weights to balance task reward and style reward is important in getting good policies. Ensuring stability, validity, and execution of the gait transition is especially challenging when considering complex terrains such as construction sites, terrains affected by natural disasters, etc. Additionally, heavy biases in the dataset, meaning data in which instances of a specific gait significantly outnumber the instances of others, can also pose a major problem.

Regarding different platforms, it is worth thinking about adjusting the natural gait. For example, the gait of a horse is much more different than that of a dog. It may be necessary to consider combining gaits with energy consumption across different platforms since it is unlikely that a single 'natural' gait works for all of them. We probably need to consider the platform's dynamic parameters (like inertia, leg weight, trunk weight, and various motor types) and the structure (different topologies). These properties of a quadrupedal robot lead to different motion characteristics, such as maximum speed and payload weight, which could be very helpful heuristic information for applying different platforms in different scenarios such as rescue operations (which may require high speed) or transportation, which demands payload.

B. Future Work

To address the limitations previously discussed, exploring latent embeddings derived from locomotion could be a promising solution. Capturing the underlying patterns across different gaits may facilitate smoother transitions between different modes of gait. Regarding the improvement of stability, incorporating style rewards might be insufficient for a robot to achieve true animal-like motion, which is more robust and adaptive in the environment when compared to a robot. A potential avenue for improvement involves estimating real-world disturbances and then applying compliant control techniques. This strategy is proven to enhance the learned controller's energy efficiency and naturalness in robotic movements, as evidenced in recent studies [34], [35].

References

[1] M. Zare, P. M. Kebria, A. Khosravi, and S. Nahavandi, "A Survey of Imitation Learning: Algorithms, Recent Developments, and Challenges." arXiv, Sep. 05, 2023.

[2] E. Vollenweider, M. Bjelonic, V. Klemm, N. Rudin, J. Lee, and M. Hutter, "Advanced Skills through Multiple Adversarial Motion Priors in Reinforcement Learning." arXiv, Mar. 23, 2022.

[3] A. Escontrela et al., "Adversarial Motion Priors Make Good Substitutes for Complex Reward Functions." arXiv, Mar. 28, 2022.

[4] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "AMP: Adversarial Motion Priors for Stylized Physics-Based Character Control," ACM Trans. Graph., vol. 40, no. 4, pp. 1–20, Aug. 2021, doi: 10.1145/3450626.3459670.

[5] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters, "An Algorithmic Perspective on Imitation Learning".

[6] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills," ACM Trans. Graph., vol. 37, no. 4, pp. 1–14, Aug. 2018, doi: 10.1145/3197517.3201311.

[7] I. Kostrikov, K. K. Agrawal, D. Dwibedi, S. Levine, and J. Tompson, "Discriminator-Actor-Critic: Addressing Sample Inefficiency and Reward Bias in Adversarial Imitation Learning." arXiv, Oct. 15, 2018.

[8] J. Ho and S. Ermon, "Generative Adversarial Imitation Learning." arXiv, Jun. 10, 2016.

[9] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation Learning: A Survey of Learning Methods," ACM Comput. Surv., vol. 50, no. 2, pp. 1–35, Mar. 2018, doi: 10.1145/3054912.

[10] P. Florence et al., "Implicit Behavioral Cloning." arXiv, Aug. 31, 2021.

[11] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved Training of Wasserstein GANs." arXiv, Dec. 25, 2017.

[12] J. Hwangbo et al., "Learning agile and dynamic motor skills for legged robots," Sci. Robot., vol. 4, no. 26, p. eaau5872, Jan. 2019, doi: 10.1126/scirobotics.aau5872.

[13] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine, "Learning Agile Robotic Locomotion Skills by Imitating Animals." arXiv, Jul. 20, 2020.

[14] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning Quadrupedal Locomotion over Challenging Terrain," Sci. Robot., vol. 5, no. 47, p. eabc5986, Oct. 2020, doi: 10.1126/scirobotics.abc5986.

[15] Y. Wang, Z. Jiang, and J. Chen, "Learning Robust, Agile, Natural Legged Locomotion Skills in the Wild." arXiv, Oct. 06, 2023.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms." arXiv, Aug. 28, 2017.

[17] L. Blondé and A. Kalousis, "Sample-Efficient Imitation Learning via Generative Adversarial Nets." arXiv, Mar. 08, 2019.

[18] M. Orsini et al., "What Matters for Adversarial Imitation Learning?" arXiv, Jun. 01, 2021.

[19] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme Parkour with Legged Robots".

[20] Z. Fu, A. Kumar, J. Malik, and D. Pathak, "Minimizing Energy Consumption Leads to the Emergence of Gaits in Legged Robots." arXiv, Oct. 25, 2021.

[21] L. Smith et al., "Learning and Adapting Agile Locomotion Skills by Transferring Experience." arXiv, Apr. 19, 2023.

[22] L. Smith, I. Kostrikov, and S. Levine, "Demonstrating A Walk in the Park: Learning to Walk in 20 Minutes With Model-Free Reinforcement Learning," in Robotics: Science and Systems XIX, Robotics: Science and Systems Foundation, Jul. 2023. doi: 10.15607/RSS.2023.XIX.056.

[23] J. Long, Z. Wang, Q. Li, J. Gao, L. Cao, and J. Pang, "Hybrid Internal Model: Learning Agile Legged Locomotion with Simulated Robot Response." arXiv, Jan. 01, 2024.

[24] M. Hutter et al., "ANYmal - a highly mobile and dynamic quadrupedal robot," in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, South Korea: IEEE, Oct. 2016, pp. 38–44. doi: 10.1109/IROS.2016.7758092.

[25] Unitree Robotics: https://m.unitree.com/

[26] B. Katz, J. D. Carlo, and S. Kim, "Mini Cheetah: A Platform for Pushing the Limits of Dynamic Quadruped Control," in 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada: IEEE, May 2019, pp. 6295–6301. doi: 10.1109/ICRA.2019.8793865.

[27] Spot: https://bostondynamics.com/products/spot/
[28] M. Shafiee, G. Bellegarda, and A. Ijspeert, "DeepTransition: Viability Leads to the Emergence of Gait Transitions in Learning Anticipatory Quadrupedal Locomotion Skills." arXiv, Jun. 14, 2023.
[29] D. Kang, S. Zimmermann, and S. Coros, "Animal Gaits on Quadrupedal Robots Using Motion Matching and Model-Based Control," in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic: IEEE, Sep. 2021, pp. 8500–8507. doi: 10.1109/IROS51168.2021.9635838.
[30] D. Kang, F. De Vincenti, N. C. Adami, and S. Coros, "Animal Motions on Legged Robots Using Nonlinear Model Predictive Control," in 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan: IEEE, Oct. 2022, pp. 11955–11962. doi: 10.1109/IROS47612.2022.9981945.
[31] D. Kang, J. Cheng, M. Zamora, F. Zargarbashi, and S. Coros, "RL + Model-based Control: Using On-demand Optimal Control to Learn Versatile Legged Locomotion," IEEE Robot. Autom. Lett., vol. 8, no. 10, pp. 6619–6626, Oct. 2023, doi: 10.1109/LRA.2023.3307008.
[32] H. Y. Ling, F. Zinno, G. Cheng, and M. van de Panne, "Character Controllers Using Motion VAEs," ACM Trans. Graph., vol. 39, no. 4, Aug. 2020, doi: 10.1145/3386569.3392422.
[33] M. Gleicher, "Retargetting motion to new characters," in Proceedings of the 25th annual conference on Computer graphics and interactive techniques - SIGGRAPH '98, Not Known: ACM Press, 1998, pp. 33–42. doi: 10.1145/280814.280820.
[34] A. Hartmann, D. Kang, F. Zargarbashi, M. Zamora, and S. Coros, "Deep Compliant Control for Legged Robots".
[35] S. Lee, P. S. Chang, and J. Lee, "Deep Compliant Control," in Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings, Vancouver BC Canada: ACM, Aug. 2022, pp. 1–9. doi: 10.1145/3528233.3530719.