Learning Robust, Agile, Natural Legged Locomotion Skills in the Wild

Anonymous Author(s)
Affiliation
Address
email

Abstract: Recently, reinforcement learning has become a promising and polular solution for robot legged locomotion. However, the corresponding learned gaits are in general overly conservative and unatural. In this paper, we propose a new framework for learning robust, agile and natural legged locomotion skills over challenging terrain. We incorporate an adversarial training branch based on real animal locomotion data upon a teacher-student training pipeline for robust sim-to-real transfer. Empirical results on both simulation and real world of a quadruped robot demonstrate that our proposed algorithm enables robustly traversing challenging terrains such as stairs, rocky ground and slippery floor with only proprioceptive perception. Meanwhile, using diverse gait patterns, the gaits are more agile, natural, and energy efficient compared to the baselines. Both qualitative and quantitative results are presented in this paper. Videos are at: https://sites.google.com/view/adaptive-multiskill-locomotion.

1 Introduction

While sim-to-real reinforcement learning exhibits robust legged locomotion skills with appealing properties, in practice, directly optimizing a task reward can lead to policies that produce behaviors undesirable to be applied in real robots, such as unnatural gaits, large contact forces, and high energy consumption. To address these challenges, previous studies have primarily employed intricate reward functions that penalize undesirable behaviors while promoting specific gait patterns[1]. Nevertheless, the process of reward engineering is laborious, and the resulting gaits still frequently appear unnatural.

To address the challenges posed by reward engineering and to achieve more natural gaits, adversarial motion priors (AMP) [2] a promising approach which leverages motion capture data and utilizes adversarial imitation learning to acquire locomotion tasks that closely resemble real-world motion data. While such method has demonstrated successful transfer from simulation to a real quadrupedal robot [3], the learned control policy is limited to traversing flat terrain in a laboratory environment, thereby lacking the capability to handle challenging terrains such as stairs or slippery ground. An intuitive extension is to train the control policy in simulation environments that incorporate different types of terrain. However, based on our experiment results, policies trained with this approach fail to achieve satisfied rewards even within simulation.

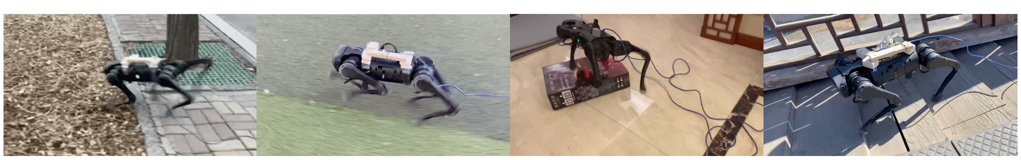

Figure 1: Experiments in real world.

In this paper, we propose a new framework which enables learning not only robust, but also agile and natural legged locomotion skills over challenging terrains in the wild. We incorporate an adver-

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

sarial training branch based on real animal locomotion data upon a teacher-student training pipeline for robust sim-to-real transfer. Experiment results show that our method successfully learn legged locomotion skills to traverse challenging terrains such as stairs, rocky ground and slippery floor.

In summary, our contributions are as follows:

- We present a framework that empower the robot with robustness and naturalness to move in the wild. The learned policy is able to adaptively transit different gaits.
- To the best of our knowledge, this is the first learning-based method enabling quadrupedal robots to gallop in the wild.

2 Method

The proposed approach comprises several building blocks which mainly support robust sim-to-real learning as well as natural gait learning from motion capture reference. An overview of the proposed framework is shown in Figure 2. We first have a phase 1 training process, which learns a teacher policy using both proprioceptive observation and the privileged information. An adversarial training process is running simultaneously to enforce agile and natural gait from motion capture reference data. Then at the phase 2 training process, we learn a student policy which takes the historical proprioceptive observations and output the final actions with the policy. This policy are directly deployed to the real robot which bridges the sim-to-real gap. In this section, we will introduce the details of each component.

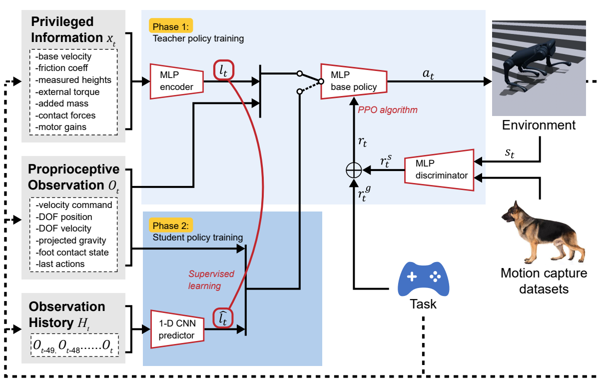

Figure 2: Overview of the traning and control framework.

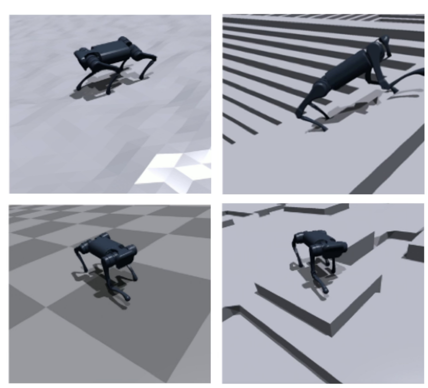

Figure 3: Terrains in simulation.

Figure 4: Transition from the 'pace' gait(frame 12) to the 'trot' gait(frame 345), and eventually to the 'gallop' gait(frame 678).

2.1 Robust Sim-to-Real Locomotion Learning

Teacher-Student Training Framework: Inspired by previous works for robust legged locomotion learning [4, 5], we integrate the teacher-student training paradigm into our framework. The teacher policy includes encoding privileged information of the environments and the robot from the simulation, while the student policy only takes observations directly available from sensors on the real robot. See appendix for more implementation details.

1. **Teacher Policy Training** In our work, the state s is composed of both the proprioceptive observation O_t and a latent vector l_t. l_t contains encoded privileged information using an encoder $l_t = \mu(x_t)$. Then a base policy π maps the concatenated state $s_t = (l_t, O_t)$ to the action command a_t. μ and π are trained jointly using PPO[6].

2. **Student Policy Training** Since the privileged information is hard to obtain in real world, we train another encoder $\hat{\mu}$ (named 'predictor'), which takes a series of historical proprioceptive observations $O_{t-T}, ..., O_{t-1}, O_t$ as inputs. The predictor is trained using supervised learning to minimize the error between the predictor output $\hat{l}_t$ and the ground truth latent l_t: $||\hat{l}_t - l_t||^2$. After obtaining the latent $\hat{l}_t$, we use the same base policy π with the teacher policy to compute the action a_t.

Enhancing Sim-to-Real Transfer: Upon the teacher-student training paradigm, we also incorporate several important techniques to enhance the sim-to-real transfer performance. See appendix for more implementation details.

- **Noise and Domain Randomization:** we incorporate observation noise to account for hardware sensor inaccuracies and transmission delays. We add randomization to physical factors and add perturbations to the robot to reduce the sim-to-real gap and enhance robustness.
- **Terrain Curriculum:** similar to [1] we generate four terrain types with varying difficulty level: plane ground, uniform noise, discrete obstacles and stairs. We also adopt the game-inspired terrain curriculum.
- **Action Filtering:** We apply a low-pass filter to the output actions which could smooth the motions and enable better sim-to-real transfer.

2.2 Natural Gait Learning with Motion Capture Reference

We hope the learned locomotion skills to be not only robust, but also natural and agile just like real animals. Inspired by adversarial motion priors (AMP) [2], we incorporate an adversarial motion style matching process into our framework, in order to learn robust, agile, and natural legged locomotion skills. See appendix for implementation details.

Motion Capture Data Reference: We utilize high-quality dog motion capture dataset provided by Zhang et al. [7]. To adapt the dog motion data to our robot, we apply inverse kinematics for motion retargeting as employed in Peng et al. [8]. Furthermore, we enhance the motion capture data by mirroring the dataset. We find that this is crucial for the sim-to-real transfer of gallop gait.

Adversarial Motion Style Matching: In order to learn agile and natural gaits, our designed reward for the reinforcement learning problem consists of both a "task" reward r_t^g and a "style" reward r_t^s. The overall reward function is given by $r_t = \omega^g r_t^g + \omega^s r_t^s$. The task reward consists of a linear velocity command tracking reward and an angular velocity command tracking reward.

The style reward is generated by a discriminator D_ϕ, which is trained to classify whether the given state transition samples are from the motion capture dataset or from the policy rollouts.

3 Experiments

We use Isaac Gym [9] simulator for training and use Unitree A1 as our robot platform in both simulation and real world. We compare the performance of our approach with two baselines:

- **Complex rewards**: Policy trained with typical model-free RL method using complex hand-designed reward function as in [1].
- **AMP**: Policy trained using adversarial motion priors as style reward to learn agile and natural legged locomotion skills [3].

We conduct both simulation and real world experiments to evaluate out method, which demonstrate that our method outperforms baselines by learning robust, agile, natural and energy-efficient legged locomotion skills.

3.1 Simulation Experiments

Experimental setup:In simulation experiments, we compare our approach's command tracking accuracy (reflected by the velocity command tracking reward) with the baselines. The experiments were conducted on three kinds of challenging terrains (as shown in Fig 3): stairs with step height of 14cm; ground randomly placed with discrete obstacles; uneven ground generated by adding uniform noise to the terrain heights.

Results: Results of the evaluation metrics in simulations are shown in Table 1. We perform 1000 independent experiments per policy using three distinct random seed-trained policies, reporting the average value and a 95% confidence interval. AMP fails to traverse stairs and discrete obstacles, while Complex Rewards fails to traverse discrete obstacles, so they are omitted in the table.

Table 1: Comparison of Command Tracking Reward

Cmd velocity	Uniform noise			Stairs		Discrete obstacles
	Complex	AMP	Ours	Complex	Ours	Ours
0.5 m/s	55.56±5.73	46.37±3.21	**62.5±0.91**	20.34±1.87	**54.99±1.21**	**57.84±1.06**
1.0 m/s	53.49±7.83	45.39±2.09	**62.55±0.65**	24.86±2.74	**50.45±1.12**	**54.24±2.77**
1.5 m/s	47.98±3.28	39.08±4.67	**62.01±0.20**	\	**30.28±2.75**	**40.64±3.67**
2.0 m/s	**59.39±7.34**	25.37±3.34	54.89±1.47	\	**11.01±1.04**	**24.80±3.92**
2.5 m/s	44.50±6.54	9.34±5.98	**53.85±1.48**	\	\	\

We can see that our controller can traverse a greater variety of complex terrains with higher command tracking reward, this might be due to the diverse motion capture data that enables the robot to switch to the most suitable gait or blend different gaits at different terrains and speeds (see Fig.4). Meanwhile, the teacher-student training architecture plays an important role in state estimation and system identification.

3.2 Real World Experiments

Experimental setup: We compare our approach's real-world performance with baselines on the following metrics: TTF: time to fall normalized by a threshold time; success rate: the ratio of the number of experiments without falling to the total number of experiments conducted; Distance: the distance the robot covers within the threshold time is normalized by the desired distance. If the robot reaches the desired distance within this time, it's set to 1.

We evaluate the performance of controllers on four types of terrain. Sample outcomes are shown in Figure 1, videos can be found on the website.

Table 2: Results of Real World Experiments

	Success rate			TTF			Distance		
	Complex	AMP	Ours	Complex	AMP	Ours	Complex	AMP	Ours
13-cm step	0.2	0	**0.8**	**1**	**1**	0.8	0.36	0.2	**0.84**
Grassland	0.8	0	**1**	1	0.1	**1**	0.92	0.1	**1**
Slippery ground	0	0.2	**0.8**	0.4	0.6	**0.9**	0.4	0.68	**0.94**
Staircase	0	0	**1**	0.98	0.68	**1**	0.56	0.1	**1**

Results: The quantitative results are shown in Table 2, each data is an average over 10 experiments. Note that the real world terrains are even more complex and diverse than that in simulation, with many unknown physical factors. Therefore, conducting real world experiments places high demands on the robustness of the controllers.

Moreover, applying a low-pass filter to the output actions significantly enhances motion smoothness, leading to notable energy efficiency improvements. We performed ablation studies on the low-pass filter's impact on energy efficiency. As depicted in Table 2, our approach demonstrates superior energy efficiency across varying velocity commands.

Table 3: Comparison of Energy Efficiency

Cmd velocity[m/s]	Avg power w/o filter [w]			Avg power w/ filter [w]		
	Complex	AMP	Ours	Complex	AMP	Ours
0.5	25.88	20.66	13.21	24.39	15.62	**11.88**
1	48.53	59.63	56.32	41.03	33.85	**33.13**

References

[1] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100. PMLR, 2022.

[2] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (TOG)*, 40(4): 1–20, 2021.

[3] A. Escontrela, X. B. Peng, W. Yu, T. Zhang, A. Iscen, K. Goldberg, and P. Abbeel. Adversarial motion priors make good substitutes for complex reward functions. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 25–32. IEEE, 2022.

[4] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

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

[6] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[7] H. Zhang, S. Starke, T. Komura, and J. Saito. Mode-adaptive neural networks for quadruped motion control. *ACM Transactions on Graphics (TOG)*, 37(4):1–11, 2018.

[8] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine. Learning agile robotic locomotion skills by imitating animals. *arXiv preprint arXiv:2004.00784*, 2020.

[9] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State. Isaac gym: High performance gpu-based physics simulation for robot learning, 2021.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

[11] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018.

[12] A. Kumar, Z. Li, J. Zeng, D. Pathak, K. Sreenath, and J. Malik. Adapting rapid motor adaptation for bipedal robots. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1161–1168. IEEE, 2022.

[13] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang. Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers. *arXiv preprint arXiv:2107.03996*, 2021.

[14] 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*, 7(2):4630–4637, 2022.

[15] A. Sharma, M. Ahn, S. Levine, V. Kumar, K. Hausman, and S. Gu. Emergent real-world robotic skills via unsupervised off-policy reinforcement learning. *arXiv preprint arXiv:2004.12974*, 2020.

[16] Z. Xie, X. Da, M. Van de Panne, B. Babich, and A. Garg. Dynamics randomization revisited: A case study for quadrupedal locomotion. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4955–4961. IEEE, 2021.

[17] S. Bohez, S. Tunyasuvunakool, P. Brakel, F. Sadeghi, L. Hasenclever, Y. Tassa, E. Parisotto, J. Humplik, T. Haarnoja, R. Hafner, et al. Imitate and repurpose: Learning reusable robot movement skills from human and animal behaviors. *arXiv preprint arXiv:2203.17138*, 2022.

[18] G. B. Margolis, T. Chen, K. Paigwar, X. Fu, D. Kim, S. Kim, and P. Agrawal. Learning to jump from pixels. *arXiv preprint arXiv:2110.15344*, 2021.

[19] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022.

[20] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal. Rapid locomotion via reinforcement learning. *arXiv preprint arXiv:2205.02824*, 2022.

[21] P. Wu, A. Escontrela, D. Hafner, K. Goldberg, and P. Abbeel. Daydreamer: World models for physical robot learning. *arXiv preprint arXiv:2206.14176*, 2022.

[22] L. Smith, I. Kostrikov, and S. Levine. A walk in the park: Learning to walk in 20 minutes with model-free reinforcement learning. *arXiv preprint arXiv:2208.07860*, 2022.

[23] Z. Xie, P. Clary, J. Dao, P. Morais, J. Hurst, and M. van de Panne. Iterative reinforcement learning based design of dynamic locomotion skills for cassie. *arXiv preprint arXiv:1903.09537*, 2019.

[24] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst. Blind bipedal stair traversal via sim-to-real reinforcement learning. *arXiv preprint arXiv:2105.08328*, 2021.

[25] J. Siekmann, Y. Godse, A. Fern, and J. Hurst. Sim-to-real learning of all common bipedal gaits via periodic reward composition. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7309–7315. IEEE, 2021.

[26] N. S. Pollard, J. K. Hodgins, M. J. Riley, and C. G. Atkeson. Adapting human motion for the control of a humanoid robot. In *Proceedings 2002 IEEE international conference on robotics and automation (Cat. No. 02CH37292)*, volume 2, pages 1390–1397. IEEE, 2002.

[27] D. B. Grimes, R. Chalodhorn, and R. P. Rao. Dynamic imitation in a humanoid robot through nonparametric probabilistic inference. In *Robotics: science and systems*, pages 199–206. Cambridge, MA, 2006.

[28] W. Suleiman, E. Yoshida, F. Kanehiro, J.-P. Laumond, and A. Monin. On human motion imitation by humanoid robot. In *2008 IEEE International conference on robotics and automation*, pages 2697–2704. IEEE, 2008.

[29] K. Yamane, S. O. Anderson, and J. K. Hodgins. Controlling humanoid robots with human motion data: Experimental validation. In *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pages 504–510. IEEE, 2010.

[30] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)*, 41(4):1–17, 2022.

A Implementation Details

A.1 State and Action Spaces

The output action a_t comprises a 12-dim target joint angle vector. The observation o_t is a 46-dim vector containing the 3-dim velocity command, 12-dim joint positions, 12-dim joint velocities, 3-dim projected gravity, 4-dim binary foot-contact states, and 12-dim last actions. The privileged information x_t is a 233-dim vector that includes the linear and angular velocity in the base frame (6-dim), friction coefficient, measured heights of some surrounding points (187-dim), external torque applied to the base (2-dim), stiffness and damping of each motor (24-dim), added mass to the base, and foot contact forces (4-dim). The encoder takes x_t as input, while the predictor takes the history observation $o_{t-T}, ..., o_t$ as input, where $T = 50$.

In order to train and conduct inference on the discriminator, we introduce the AMP observation denoted as s_t, which is comprised of joint positions, joint velocities, foot positions in base frame, base linear velocities, base angular velocities, and base height, resulting in a 67-dimensional vector. The input provided to the discriminator consists of the state transition (s_{t-1}, s_t).

A.2 Network Architecture

The teacher encoder is a 2-layer multi-layer perceptron (MLP) that takes the privileged information $x_t \in \mathbb{R}^{233}$ as input and outputs the latent vector $z_t \in \mathbb{R}^8$. The hidden layers have dimensions [256,128].

The base policy is a 3-layer multi-layer perceptron (MLP) that takes the current observation $o_t \in \mathbb{R}^{46}$ and the latent vector z_t as input and generates a 12-dimensional target joint angle output. The hidden layers have dimensions [512, 256, 128].

The student predictor begins by encoding each observation from recent steps into a 32-dimensional representation. Next, a one-dimensional convolutional neural network (1-D CNN) convolves these representations along the time dimension. The layer configurations, such as input channel number, output channel number, kernel size, and stride, are set to [32, 32, 8, 4], [32, 32, 5, 1], and [32, 32, 5, 1]. The flattened output from the CNN is then passed through a linear layer to predict $\hat{z}_t$.

The discriminator employs an MLP with hidden layers of size [1024, 512].

A.3 Adversarial Motion Style Matching

The overall reward function is given by $r_t = \omega^g r_t^g + \omega^s r_t^s$. The ratio of these two part is quite critical to the robot's performance. In this work, we chose ω^g to be 0.35, while $\omega^s = 0.65$. The task reward is defined based on the specific task we aim to accomplish, here it consists of a linear velocity command tracking reward and an angular velocity command tracking reward:

$$r_t^g = w^v \exp\left(-\left|\hat{v}_t^{\text{xy}} - v_t^{\text{xy}}\right|\right) + w^\omega \exp\left(-\left|\hat{\omega}_t^z - \omega_t^z\right|\right) \tag{1}$$

where w^v, w^ω, and w^τ are the coefficients. $\hat{v}_t^{\text{xy}}$ and $\hat{\omega}_t^z$ represent the linear and angular velocity commands, respectively. To ensure robustness and learn diverse gait patterns, different ranges of velocity commands are defined for each terrain type, as listed in A.6. The velocity commands are randomly sampled from the specified ranges.

The style reward is generated by a discriminator D_ϕ, which is trained to classify whether the given state transition samples are from the motion capture dataset or from the policy rollouts, where ϕ denotes the discriminator's parameters. The optimization objective of the discriminator is as follows:

$$\begin{aligned} \arg\min_\phi \; & \mathbb{E}_{(s,s')\sim\mathcal{D}}\left[\left(D_\phi\left(s,s'\right)-1\right)^2\right] \\ & + \mathbb{E}_{(s,s')\sim\pi_\theta(s,a)}\left[\left(D_\phi\left(s,s'\right)+1\right)^2\right] \\ & + \frac{w^{\text{gp}}}{2}\mathbb{E}_{(s,s')\sim\mathcal{D}}\left[\left\|\nabla_\phi D_\phi\left(s,s'\right)\right\|^2\right], \end{aligned} \tag{2}$$

where $\mathcal{D}$ denotes the motion capture dataset, The first two terms incentivize the descriminator to output 1 for transition pairs from the mo-cap dataset, while output -1 for transition pairs from the policy rollouts. ω^{gp} is the coefficient for gradient penalty which reduces oscillations in the adversarial training process. The style reward is then defined as:

$$r_t^s(s_t, s_{t+1}) = \max\left\{0, 1 - 0.25(D_\phi(s_t, s_{t+1}) - 1)^2\right\} \tag{3}$$

Therefore, the policy is trained through reinforcement learning to maximize the reward function as a generator, while the discriminator is trained using both the motion dataset $\mathcal{D}$ and the data generated during policy rollouts, forming an adversarial motion style matching framework.

A.4 Learning Algorithm

We utilized Proximal Policy Optimization (PPO) as the reinforcement learning algorithm to train both the base policy and teacher encoder concurrently. The training process was composed of 50,000 iterations, with each iteration involving the collection of a batch of 131,520 state transitions. These transitions were evenly divided into 4 mini-batches for processing. To maintain a desired KL divergence of $KL^{desired} = 0.01$, we automatically tuned the learning rate using the adaptive LR scheme proposed by [10]. The PPO clip threshold was set to 0.2. For the generalized advantage estimation [6], we set the discount factor γ to 0.99 and the parameter λ to 0.95.

To optimize the objective defined in Eq (2), we trained the discriminator using supervised learning. We set the gradient penalty weight to $w^{\text{gp}} = 10$. The style reward weight is $w^s = 0.65$ and the task reward weight is $w^g = 0.35$.

The student encoder was trained with supervised learning, minimizing the mean squared error (MSE) loss between the latent vector z_t output by the teacher encoder and the predicted latent vector $\hat{z}_t$ output by the student encoder.

Throughout all training phases, we utilized the Adam optimizer with β values set to (0.9, 0.999), and ϵ set to $1e - 8$.

A.5 Terrain Curriculum

We utilize four types of terrains: plane ground, uniform noise, discrete obstacles, and stairs. Before proceeding to a more challenging type of terrain, the robot needs to successfully traverse the current terrain and achieve a satisfied task reward. The threshold we use to increase terrain difficulty consists: (1)The robot successfully crosses the center of a terrain block within a single episode; (2)The linear velocity tracking reward surpasses 80% of the maximum achievable reward which corresponds to 'perfectly' accurate tracking.

In contrast, the robots are reset to easier terrains if they fail to travel more than half of the distance required by their command linear velocity within an episode. This adaptive curriculum mechanism enables us to stably learn robust locomotion skills for the robot.

A.6 Command Range

Table 4: Command Ranges for Different Terrains

	plane ground	stairs	discrete obstacles	uniform noise
lin vel cmd (m/s)	[-1.0,3.0]	[0,1.6]	[0,1.6]	[-1.0,2.5]
ang vel cmd (rad/s)	[-1.5,1.5]	[-1.0,1.0]	[-1.0,1.0]	[-1.5,1.5]

A.7 Noise and Domain Randomization

Table 5: Ranges of Randomization and Perturbations

environmental randomization	friction coefficient	[0.25,1.5]
	added mass	[-1.0,1.0]kg
	motor gain multiplier	[0.85,1.15]
perturbation	external torque	[-3.0,3.0]Nm
	linear velocity perturbation	[-1.0,1.0]m/s
	angular velocity perturbation	[-3.0,3.0]rad/s
sensor noise	joint position	[-0.03,0.03]rad
	joint velocity	[-1.5,1.5]rad/s
	base linear velocity	[-0.1,0.1]m/s
	base angular velocity	[-0.3,0.3]m/s
	gravity	[-0.49,0.49]m^2/s
	height measurement	[-0.01,0.01]m

B RELATED WORK

B.1 Reinforcement Learning for Legged Locomotion

Recent advancements in deep reinforcement learning for legged locomotion have demonstrated its promising future. Lee et al. [4] applied teacher-student training to the quadruped robot ANYmal, resulting in a robust controller capable of traversing challenging terrains, which is similar to the teacher-student training paradigm as ours. Peng et al. [8] introduced the use of Deep Mimic [11] to learn robotic locomotion skills by imitating animals. We adopted the similar motion retargeting technique as [8]. Similar to [4], Kumar et al. [5] trained locomotion policies with rapid motor adaptation, enabling them to quickly adapt to environmental changes. Building upon this, Kumar et al. [12] extended the RMA algorithm to the bipedal robot Cassie. Yang et al. [13] employed a cross-modal transformer to learn an end-to-end controller for quadrupedal navigation in complex environments. Ji et al. [14] trained a neural network state estimator to estimate robot states that cannot be directly inferred from sensory data. Escontrela et al. [3] utilized Adversarial Motion Priors (AMP) to train control policies for a quadrupedal robot, highlighting that AMP can effectively substitute complex reward functions. The relationship between [3] and ours is that ours further adapted the AMP algorithm to work on challenging terrains. Sharma et al. [15] trained a reinforcement learning controller using unsupervised skill discovery and successfully transferred it to a real quadruped robot. Xie et al. [16] revisited the necessity of dynamics randomization in legged locomotion and provided suggestions on when and how to employ dynamics randomization. Bohez et al. [17] trained a low-level locomotion controller for a quadruped robot by imitating real animal data, utilizing this controller to accomplish various tasks. Margolis et al. [18] trained policies to perform jumps from pixel inputs, while Miki et al. [19] trained a locomotion controller using observations of the height map of the terrain around the robot's base. Rudin et al. [1] employed massively parallel simulation environments to significantly accelerate the training process of a locomotion controller. Margolis et al. [20] trained a locomotion controller for the Mini Cheetah robot, enabling it to achieve speeds of up to 3.9m/s, surpassing traditional controllers' speeds by a large margin. Other notable works include directly learning locomotion skills in the real world [21, 22], as well as learning locomotion skills for bipedal robots [23, 24, 25].

B.2 Motion Control from Real World Motion Data

Imitating a reference motion dataset offers an approach to designing controllers for skills that are challenging to manually encode. Pollard et al. [26, 27, 28, 29] employ motion tracking techniques, where characters explicitly mimic the sequence of poses from reference trajectories. Learning from real-world motion provides an alternative to crafting complex rewards for synthesizing natural motion. Peng et al. [11] adapt reinforcement learning (RL) methods to learn robust control policies capable of imitating a wide range of example motion clips. Leveraging GAN-style training, Peng et al. [2] learn a ”style” reward from a reference motion dataset to control the character’s low-level movements, while allowing users to specify high-level task objectives. Escontrela et al. [3] utilize the framework proposed by Peng et al. [2] to train a locomotion policy for a quadrupedal robot to traverse flat ground. Additionally, Peng et al. [30] present a scalable adversarial imitation learning framework that enables physically simulated characters to acquire a wide repertoire of motor skills, which can be subsequently utilized to perform various downstream tasks.