Motion Priors Reimagined: Adapting Flat-Terrain Skills for Complex Quadruped Mobility

Zewei Zhang
Department of Mechanical Engineering, EPFL

Chenhao Li
ETH AI Center, ETH Zurich

Takahiro Miki
Robotic Systems Lab, ETH Zurich

Marco Hutter
Robotic Systems Lab, ETH Zurich

Abstract: Reinforcement learning (RL)-based motion imitation methods trained on demonstration data can effectively learn natural and expressive motions with minimal reward engineering but often struggle to generalize to novel environments. We address this by proposing a hierarchical RL framework in which a low-level policy is first pre-trained to imitate animal motions on flat ground, thereby establishing motion priors. A subsequent high-level, goal-conditioned policy then builds on these priors, learning residual corrections that enable perceptive locomotion, local obstacle avoidance, and goal-directed navigation across diverse and rugged terrains. Simulation experiments illustrate the effectiveness of learned residuals in adapting to progressively challenging uneven terrains while still preserving the locomotion characteristics provided by the motion priors. Furthermore, our results demonstrate improvements in motion regularization over baseline models trained without motion priors under similar reward setups. Real-world experiments with an ANYmal-D quadruped robot confirm our policy's capability to generalize animal-like locomotion skills to complex terrains, demonstrating smooth and efficient locomotion and local navigation performance amidst challenging terrains with obstacles.

Keywords: Motion Prior, Reinforcement Learning, Locomotion, Local Navigation

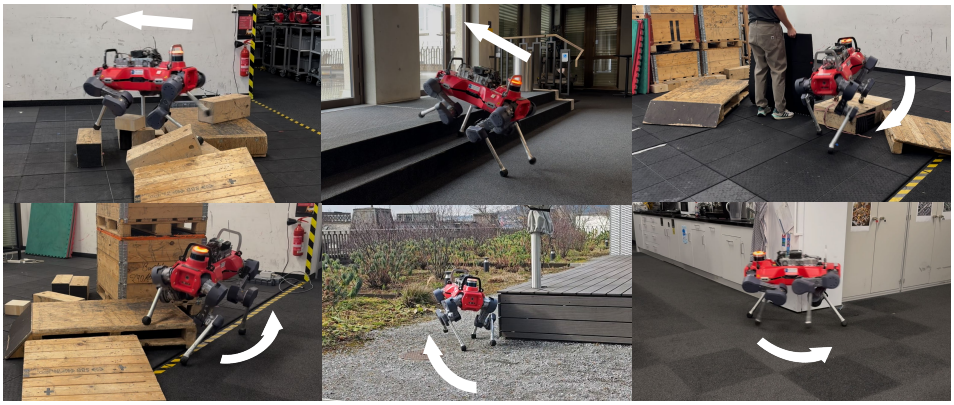

Figure 1: ANYmal-D hardware experiments in indoor/outdoor terrains (stairs, random blocks, high obstacles). The white arrow marks the direction of movement toward the specified goal position. The robot employs an animal-like gait to traverse uneven ground and avoid obstacles, demonstrating our policy's transfer of flat-ground motion priors to complex environments. (See supplementary videos at `https://anymalprior.github.io/`)

1 Introduction

Legged locomotion remains one of the most challenging problems in robotics, and reinforcement learning (RL) has recently emerged as a promising approach to tackle this complexity. Contemporary RL locomotion controllers have successfully enabled robots to achieve smooth and stable motion while accurately tracking base velocity [1, 2, 3] or goal position commands [4, 5, 6] across various

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

terrains. Additionally, position-based locomotion policies have demonstrated the capability to perform effective local navigation without relying on an external high-level planning module. Despite these advancements and their successful deployment on hardware platforms, both velocity-based and position-based locomotion controllers heavily depend on meticulous reward design and struggle to learn expressive, animal- or human-like motions.

Motion imitation has emerged as an efficient alternative to alleviate these challenges by guiding RL with reference trajectories obtained from teleoperation [7, 8], motion capture [9, 10], or trajectory optimization [11]. Such motion imitation-based policies have enabled natural and agile maneuvers, such as jumping [10, 12] and backflips [13], which is particularly challenging to learn from scratch due to the extensive reward design and tuning.

However, motion imitation-based method is not without its limitations. Their performance is inherently bounded by the fidelity and specificity of the reference data. When a policy is strictly forced to mimic a fixed set of demonstrations, it tends to overfit to the specific characteristics how the reference data are recorded. As a result, when deployed in an environment with different dynamics (e.g. terrain types), the policy often fails to reproduce the intended motion style consistently. This issue arises due to covariate shift, the mismatch between the distribution of states encountered during training (from the demonstration data) and those encountered in the new environment, which can lead to degraded performance and style inconsistency. Although recent works have sought to improve adaptation beyond the environment where demonstration data are collected [10, 14, 15], these approaches either rely on reference trajectories through trajectory optimization method or are confined to setups with only minimal environmental differences. Adaptation in more challenging and diverse scenarios using readily accessible, minimally processed references, such as raw retargeted animal motion data, remains underexplored.

In this work, we propose a hierarchical reinforcement learning pipeline that addresses motion imitation challenges. Our framework begins by pre-training motion priors using motion imitation-based RL on animal datasets collected exclusively from flat terrain. These priors subsequently facilitate the training of a high-level goal-reaching policy within a position-based formulation, enabling effective adaptation to complex, non-flat environments and accomplishing perceptive locomotion and local navigation. During deployment, our policy navigates to target goals while preserving the natural animal-like motion style and executing smooth obstacle avoidance behaviors.

Our contributions are: **(a)** a hierarchical RL framework that combines pre-training of motion priors from flat-terrain animal data with a task training stage that learns residual corrections for improved adaptability; **(b)** a unified pipeline achieving both perceptive locomotion and local navigation in a smooth, animal-like gait; and **(c)** a comprehensive analysis of how motion priors and learned residuals influence overall task and locomotion performance.

Simulation and hardware experiments demonstrate that our framework not only retains the training efficiency of motion imitation but also effectively generalizes to challenging, real-world locomotion and local navigation tasks.

2 Related Work

2.1 Motion Imitation with Reinforcement Learning

Motion imitation via RL has emerged as a reliable strategy for acquiring diverse motor skills directly from reference data. In legged locomotion, such approaches have been effective at transferring demonstrative behaviors to real-world systems. For instance, Peng et al. [16] extend Deepmimic framework [17] to real quadrupeds by using retargeted animal motion data and reward defined by the tracking error between the reference and actual states. Similarly, He et al. [9] have shown that humanoid robots can achieve agile, long-range movements leveraging comparable techniques. Complementary studies by Li et al. [18] and Watanabe et al. [19] further streamline the imitation pipeline by automating motion representation and phase labeling for Deepmimic-based methods.

An alternative line of research employs adversarial motion priors (AMP), where a GAN-style discriminator learns the distribution of state trajectories from demonstration data [12, 13, 20, 21].

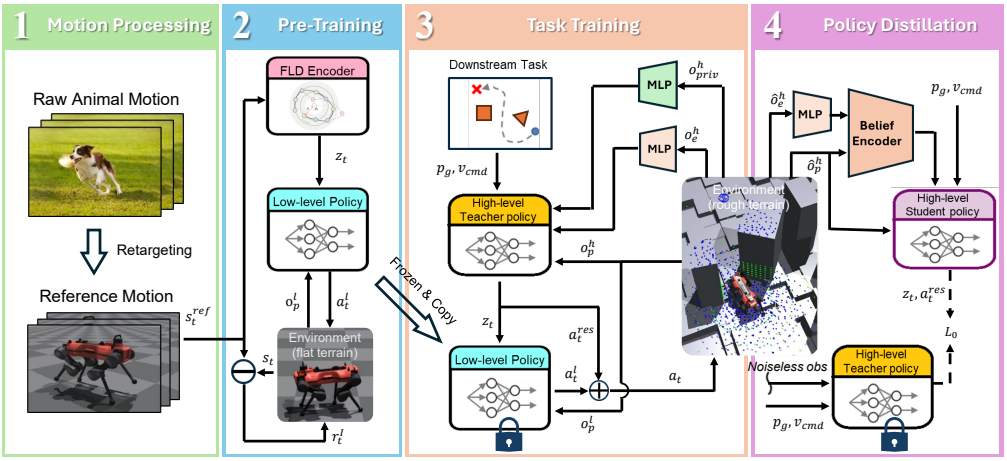

Figure 2: Overview of our training framework for quadruped locomotion and local navigation: (1) retarget animal mocap data based on the robot's configuration;(2) pre-train an FLD encoder and low-level policy on flat terrain for motion priors; (3) train a high-level teacher to output latent commands and joint residuals for rough-terrain adaptation; (4) distill into a student policy for sim-to-real using only noisy observations. Green dots: downsampled Velodyne-LiDAR scans; blue dots: elevation scans around the robot's feet.

Nevertheless, AMP-based approaches still face inherent adversarial learning challenges, including instability during discriminator training and mode collapse, especially when the dataset contains diverse motions [22]. In our work, we adopt the Deepmimic-based approach from Li et al. [18] due to its ability to yield refined motion representations and training tractability compared to the AMP-based methods.

2.2 Task Adaptation with Motion Priors

Motion imitation-based RL policies often serve as low-level motion priors that underpin more complex, high-level tasks. These motion skills can be encoded into a low-dimensional latent space and reused by high-level task policies through hierarchical conditioning [10, 23, 24]. Alternatively, some methods directly incorporate style rewards into task training [14, 21].

For example, Han et al. [10] propose a two-stage training strategy where a low-level imitation policy based on animal motions is first acquired and then used to train high-level policies that adapt to various tasks on rough terrains. Although their method successfully generates animal-like motions across multiple tasks on various terrains, it remains dependent on training with motion data from uneven terrains and on manually calibrating the simulation environment to match the specific environmental setups under which the training data are collected for the imitation performance.

In contrast, Wu et al. [14] achieve stable locomotion on complex terrains using motion priors learned solely from flat-terrain data, by integrating an AMP-based reward into the task training. However, this approach relies on carefully crafted trajectories obtained via trajectory optimization algorithms and typically captures a trotting gait. Extending the generalization of motion priors derived from raw retargetted animal data, including other gait patterns such as walking, pacing, or cantering into novel terrains, remains underexplored. Our approach addresses this gap by learning joint residuals on top of low-level priors which is only trained on flat terrain, thereby enhancing adaptability to diverse and challenging environments beyond where the motion data are recorded.

2.3 Local Navigation at Locomotion Level

In autonomous legged robotics, local navigation is commonly managed by a high-level planning module that continuously outputs waypoints [25, 26, 27] or velocity commands [28, 29] for a low-level locomotion controller. However, developing an integrated, end-to-end policy that simultaneously handles locomotion and navigation is desirable, as it leverages the full capabilities of the robot to determine optimal actions.

Prior work using position-based task rewards [6, 30] has demonstrated emergent local navigation behavior within a single learned locomotion policy. Yet, these approaches generally necessitate exten-

sive reward shaping to produce smooth motions and are predominantly applied to local navigation tasks in near-flat terrains.

Building on these developments, our aim is to improve the efficiency and performance of local navigation training at the locomotion level. By integrating motion priors derived from animal demonstrations (originally collected on flat ground) into a position-based locomotion framework, our pipeline produces a smooth, animal-like gait that empowers the robot to traverse varied terrains and efficiently circumvent local obstacles en route to its goal.

3 Method

In this section, we describe our hierarchical RL training framework, which is designed to harness the benefits of motion imitation and train a more powerful high-level policy to extend the capabilities of the original motions only feasible on flat ground for the acquisition of locomotion, local obstacle avoidance, and goal-reaching navigation skills on more challenging terrains.

As illustrated in Fig. 2, our approach is divided into four phases: motion processing, pre-training, task training and policy distillation for sim2real. Prior to pre-training, animal motion data are retargetted to align with the robot's configuration. During pre-training, we first train the FLD encoder on offline data and subsequently learn a low-level policy using the extracted latent representations as the input to generate physical motions. In the task training phase, the frozen low-level policy acts as a motion prior, while we train a high-level teacher policy with privileged information to handle complex tasks over diverse terrains by outputting latent command to the low-level policy and augmenting the low-level skills with residuals. Lastly, the high-level teacher policy is distilled into the student policy with noisy observation to narrow the sim2real gap. The following subsections provide further details.

3.1 Motion Preprocessing

Before training the policy to mimic animal movements, the raw motion capture data need to be retargetted to match the robot's kinematic configuration. We employ inverse kinematics to convert the raw data based on the positions of the neck, pelvis, and feet, following the established pipeline in Peng et al. [16]. The selected animal mocap data, sourced from [10, 31], comprise various gait patterns, such as walking, pacing, and cantering, performed at different speeds on flat ground.

3.2 Pre-training

Given the retargeted motion data, the pre-training stage focuses on acquiring a low-level policy that mimics the demonstrated motor skills. Crucial to this process is the training of an encoder that compresses the motion trajectories into low-dimensional latent vectors, thereby enabling efficient reuse of these low-level skills in later stages.

We first pre-train the FLD model which comprises both an encoder and a decoder, to capture a representation of motion patterns before proceeding to train the low-level imitation policy, following the framework outlined in Li et al. [18]. More information on FLD training is detailed in Appx. 7.1.

With the refined motion representation in hand, we next leverage the latent embedding to train a low-level motion imitation policy. In this phase, the policy is trained on flat terrain while adhering to the FLD pipeline for motion learning. However, several modifications are introduced to boost imitation performance. Specifically, we eliminate the decoder and replace the reconstructed trajectories with ground-truth trajectories, bypassing the phase propagation mechanism entirely. During training, motion clips are randomly sampled from the dataset, and the encoder dynamically computes latent embeddings from the input reference state sequences. These modifications may help reduce performance degradation from FLD decoder reconstruction errors while still maintaining a structured latent space that captures the motion's global periodic features.

The low-level policy operates in an action space defined by 12-dimensional joint actions a_t^l, and its observation o_p^l comprises proprioceptive signals including base linear and angular velocities, projected gravity vectors, joint positions, and the latent encodings. The reward function is composed of an imitation component and a regularization component. Additional details regarding the reward functions are provided in Appx. 7.2. Consequently, the trained low-level policy that bridges the

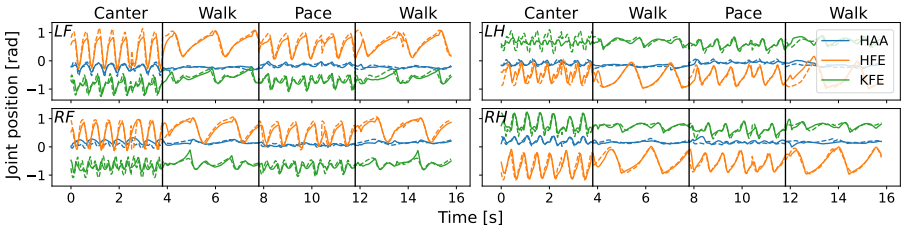

Figure 3: Actual vs. reference joint positions for the low-level policy on flat ground. Dashed lines are reference trajectories; solid lines are the actual trajectories. The plot (canter → walk → pace → walk) shows close alignment between reference and actual motions.

robot's motor skills with low-dimensional latent embeddings not only serves as the command interface for executing low-level skills, but also lays the foundation for subsequent high-level policy training.

3.3 Task Training

After training the low-level policy as a motion prior on flat-ground data, we extend its capabilities to complex environments via task training. In this phase, a high-level teacher policy is learned on top of the frozen low-level policy to address locomotion and local navigation tasks across various terrains.

The high-level teacher policy outputs 16-dimensional latent commands z_t and 12-dimensional joint residuals a_t^{res} to refine the basic motion skills. Its observation space includes noiseless proprioceptive inputs o_p^h (identical to those used by the low-level policy), noiseless exteroceptive inputs o_e^h, privileged states o_{priv}^h (capturing leg contact information, friction coefficients, and external disturbances), and task-specific inputs. For exteroception, we fuse elevation scans around each robot foot [32] with a downsampled Velodyne LiDAR scan arranged in a sparse conical pattern to provide a comprehensive environmental profile. As illustrated in Fig. 2, we further employ small MLPs to encode these terrain and privileged state inputs.

For the downstream task, our objective is to train an integrated policy that leverages motion priors to reach a goal in challenging, rough terrains including uneven surfaces, stairs, slopes, and high obstacles. The task input comprises the goal position $p_g = (p_{g,x}, p_{g,y})$, defined relative to the robot's base frame, and a velocity command v_{cmd} that coarsely regulates forward motion toward the goal. We define the task rewards as follows:

$$r_{reach} = \frac{1}{T_r}\left(1 - \frac{\|\mathbf{d}\|_2}{2}\right) \quad \text{if } t > T - T_r \text{ and } \|\mathbf{d}\|_2 < 2; \text{else } 0, \tag{1}$$

$$r_{vel} = \min(v_{cmd}, \langle \mathbf{v}, \mathbf{d} \rangle) \quad \text{if } \|\mathbf{d}\|_2 > 0.15; \text{else } 0, \tag{2}$$

where t, T, and T_r denote the current time, the interval between successive position commands, and the threshold time for reward computation, respectively. $\mathbf{d}$ denotes the displacement vector from the robot base to the target, and $\mathbf{v}$ represents the robot's base linear velocity. Since r_{reach} tends to be sparse, the additional velocity reward r_{vel} provides a denser learning signal to effectively guide local navigation in cluttered environments with high obstacles.

We maintain the same regularization rewards and weights used in the low-level policy (excluding tracking terms) and add new residual penalty terms to constrain the joint residual corrections (Eq. 3).

$$r_{res} = w_{res} \sum_{i=1}^{12} (a_{t,i}^{res})^2, \tag{3}$$

where w_{res} denotes the weight for the residual penalty. More details on the reward functions for the high-level teacher policy are provided in Appx. 7.2.

3.4 Policy Distillation for Sim2Real

To bridge the sim-to-real gap in perceptive locomotion, we employ a privileged learning strategy [33] to distill a high-performance teacher policy into a student policy. In our approach, the teacher policy is initially trained under ideal conditions using privileged, noiseless proprioceptive and exteroceptive observations. Subsequently, we distill a student policy from the teacher that operates solely on noisy

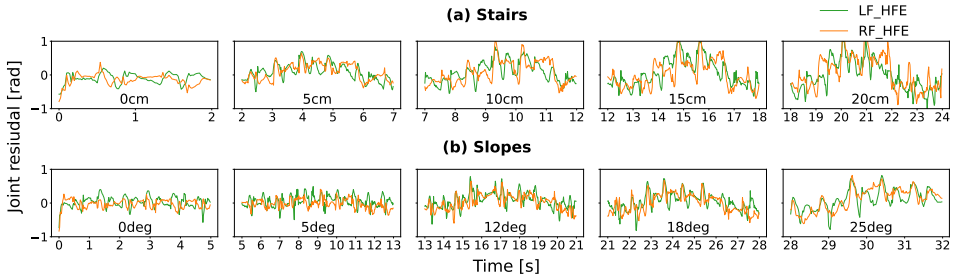

Figure 4: Comparison of high-level joint residuals for the HFE joint on the left front (LF) and right front (RF) leg across two terrain types: (a) pyramid stairs and (b) pyramid slopes with a rugged surface (see Fig. 11). Each terrain type is divided into five difficulty levels (displayed at the bottom of each subplot), with difficulty increasing from left (easy) to right (hard).

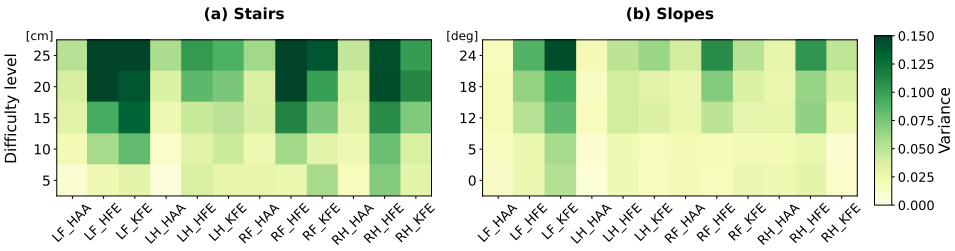

Figure 5: Variance in joint residuals of LF-HFE joint, as observed in the experiment depicted in Fig. 11. Darker colors denote higher involvement of joint residuals on specific terrain types.

proprioceptive and exteroceptive data, without access to privileged information through supervised learning. As shown in Fig. 2, a recurrent belief encoder, implemented with a Gated Recurrent Unit (GRU), is integrated into student policy to reconstruct the privileged state and recover noiseless exteroceptive signals from noisy proprioceptive and exteroceptive inputs. The design of the belief encoder follows Miki et al. [2]. The distillation process is guided by a behavior loss L_0 that quantifies the discrepancy between the teacher's and student's actions. This teacher-student framework allows us to first establish a high-performance teacher policy under controlled conditions, and then transfer that performance to a student policy that is robust under realistic, noisy operating conditions.

4 Experiments

4.1 Experiment Setup

We select ANYmal-D as our robot platform and introduce several experiments to demonstrate and verify the effectiveness of our framework. Our experiments aim to **(a)** test whether a policy enriched with latent motion priors and joint residual corrections can reliably navigate to target goals and avoid local obstacles on complex terrains; **(b)** evaluate how different penalties on residual actions affect locomotion and goal-reaching performance; and **(c)** examine the motion regularization performance improvement over a baseline RL controller trained from scratch under similar reward conditions.

4.2 Simulation Results

4.2.1 Low-level policy

We train and test a low-level policy that can demonstrate multiple animal-like motions imitated in simulation. As mentioned in Sec. 3, the low-level policy mimics motions that include walking, pacing, and cantering gaits, and each is at different forward velocities. Each skill can be performed on flat terrain and also transition smoothly between each other by commanding different latent embeddings. Figure 3 highlights the strong imitation performance of the low-level policy by comparing the reference positions with the actual positions of all joints. Figure 10 also shows the footfall sequence of individual motion skills learned in the low-level policy.

4.2.2 High-Level Policy

We test our high-level student policy in simulation under exteroceptive noise conditions that resemble real-world scenarios. The evaluation is carried out on multiple terrains, including stairs, boxes of

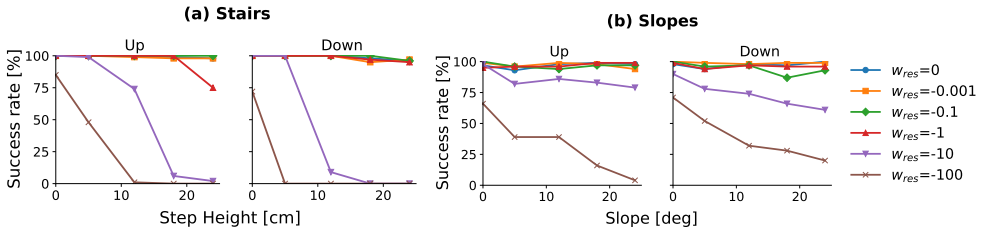

Figure 6: Goal-reaching success rate on stairs and sloped terrains across different difficulty levels, with the left subplots representing ascent and the right ones representing descent. Each terrain type is evaluated over 100 trials, with randomized initial robot poses in each experiment.

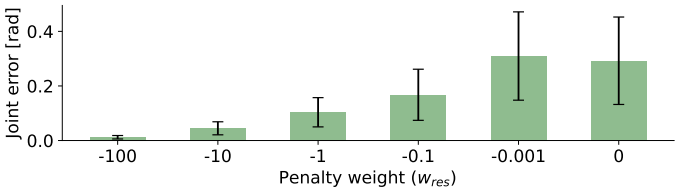

Figure 7: Average deviation of actual joint positions from reconstructed reference trajectories with FLD decoder over five seconds of forward walking on flat ground, evaluated at different residual penalty weights. Lower deviations indicate gait behavior that more closely matches the animal motions.

varying heights, slopes with rugged surfaces, and high obstacles. Examples of these terrains can be found in Appx. 7.3. In all terrain types tested, our high-level policy achieves a high success rate in reaching target positions. The policy effectively learns to generate appropriate residuals on top of the low-level motions. By integrating the animal-style smoothness derived from low-level motion priors with adaptive high-level residuals, the robot exhibits natural gait behaviors and bypassed local obstacles directly through perception, eliminating the need for a separate navigation module to output waypoints. Refer to Appx. 8.2 for a detailed analysis of the performance.

As shown in Fig. 4 and Fig. 5, we analyze the residual changes with the increasing difficulty levels of the terrains on the pyramid stairs and the pyramid slopes (Fig. 11). Based on the results in Fig. 5, we observe an increasing trend in the variance of the joint residuals in all joints as the terrain on which the robot is walking becomes more challenging, suggesting a growing contribution of residuals. For a detailed analysis, we select the joint residuals of the joint LF-HFE and RF-HFE as a representative. According to the plots, the high-level policy continuously adjusts the joint residual based on the current terrain situation. Peaks with continuously increasing height can be observed in Fig. 4 (a) and (b). The variation in the joint residual likely reflect changes in the robot's base orientation and the challenges of traversing steps of varying heights or slopes. Since the low-level motions alone cannot fully address these conditions, an adaptive residual component is needed to maintain effective locomotion. These results demonstrate that the high-level policy can successfully leverage joint residuals to generalize to non-flat and complex terrains.

4.2.3 Impact of Penalty on Residual Actions

We find that training with decreasing values of the joint residual penalty results in stronger adaptation ability but produces less-regularized motions.We train six high-level teacher policies by selecting six penalty weights of the joint residuals ranging from -100 to 0, and investigate how it would affect the performance of locomotion and task completion. As shown in Fig. 6, all policies achieve a high success rate in near-flat terrain. However, an excessively high penalty (the weight larger than -10) reduces the efficiency of tackling higher steps and steeper slopes. By contrast an overly low penalty (the weight less than -0.001) may allow residuals to grow without bound, yielding high success rate but resulting in gaits that no longer follow the regularized motions provided by the motion priors, even on flat and near-flat terrain (see Fig. 7 and supplementary videos). This illustrates a tradeoff between preserving the motion style imparted by the priors and boosting the policy's adaptability across varied terrains.

The performance of our high-level policy underscores the effective regularization achieved by the motion priors embedded in the low-level policy. Rather than engaging in extensive tuning of multiple

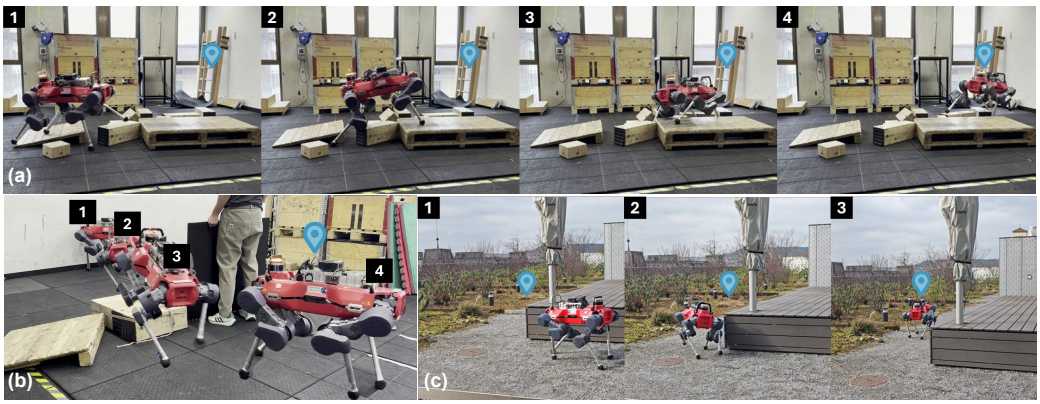

Figure 8: Real world deployment of the high-level student policy. The sequence of images illustrates the ability of the robot to reach the goal position (blue map pin) on complex terrain while avoiding the local obstacles.

reward weights, our approach hinges on a single primary penalty term that constrains the sum of the squared residuals. Our comparative experiments further confirm that including this penalty is crucial for balancing task-completion rewards with deviations from the motion priors, thereby improving the learned policy's overall performance.

4.2.4 Comparison with Baseline RL Model

To evaluate the impact of incorporating motion priors in task-specific training, we compare our high-level teacher policy with a baseline RL model that is trained entirely from scratch without leveraging any low-level motion priors. Both models are provided with the same observation space configuration except for latent encodings. However, the baseline model's action space is limited to a 12-dimensional vector of joint actions. The results display that the baseline policy, trained under the same reward setup without additional tuning and exploration techniques, tends to exhibit a jumping gait despite its ability to navigate challenging terrains (see supplementary videos). Incorporating additional regularization terms tuning on base acceleration or vertical velocity, for example, could encourage more natural locomotion. In contrast, our model, which integrates motion priors, achieves animal-like gait under identical reward conditions only by including the penalty term for joint residuals. This indicates that the learned motion priors inherently enforce the natural movement style learned from animal data, enabling us to cut back on extra regularization and streamline the tuning process.

4.3 Evaluation in Real World

In the final part, we evaluate the high-level student policy on the ANYmal-D robot in real-world environments (see Fig. 1 and Fig. 8). The test area comprises random steps, stairs, and various obstacles that the robot must avoid. With a fixed goal provided, the robot navigate to the target while maintaining a smooth animal-style gait without requiring explicit waypoint generation for obstacle bypassing. We encourage readers to refer to supplementary videos for further details. These results further validate the locomotion and local navigation capabilities of our system in real-world scenarios.

5 Conclusion

We presented a hierarchical RL framework that fused low-level animal motion priors with high-level residual learning for goal-directed locomotion across complex terrains. This design reduced reward-tuning effort for motion regularization and improved the robustness and adaptability of flat-terrain skills. In simulation, we showed that learned joint residuals achieved a tradeoff between adhering to motion priors and adapting to novel terrains, and that incorporating these priors produced more natural animal-style gaits than baseline RL controllers under similar reward conditions. Finally, hardware experiments with ANYmal-D confirmed its capability for perceptive locomotion and obstacle-aware local navigation across diverse terrains.

6 Limitations and Future Work

Despite good performance in perceptive locomotion and local navigation, our approach has several limitations. First, our high-level policy can suffer from mode collapse, habitually relying on a single low-level gait and using residuals only to adapt. Incorporating exploration strategies or diversity-driving rewards during training may alleviate this issue and encourage the policy to exploit the full range of low-level motion skills.

Second, our training scenarios remain somewhat constrained. Future work could be extending our low-level policy to a wider range of motor behaviors (e.g. jumping, crawling) and improving the adaptability for even more challenging terrains, such as gaps, stepping stones, and environments with overhanging obstacles.

In addition, although this work primarily focuses on active learning to command low-level motions for high-level tasks and demonstrates adaptation with a limited set of motions, we believe the proposed training architecture also offers an efficient and scalable framework for enhancing skill adaptation. This can be achieved by randomly sampling latent encodings of different motions during training while manually commanding fixed, specific motions for high-level task inference.

Acknowledgments

This research was supported by the ETH AI Center and the Swiss National Science Foundation through the National Centre of Competence in Automation (NCCR Automation). We also thank Hehui Zheng for her assistance with the hardware experiments.

References

[1] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019. doi:10.1126/scirobotics.aau5872.

[2] 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. doi:10.1126/scirobotics.abk2822.

[3] Z. Zhuang, S. Yao, and H. Zhao. Humanoid parkour learning. In *8th Annual Conference on Robot Learning*, 2024.

[4] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme parkour with legged robots. *arXiv preprint arXiv:2309.14341*, 2023.

[5] C. Zhang, N. Rudin, D. Hoeller, and M. Hutter. Learning agile locomotion on risky terrains, 2024.

[6] J. Ren, T. Huang, H. Wang, Z. Wang, Q. Ben, J. Pang, and P. Luo. Vb-com: Learning vision-blind composite humanoid locomotion against deficient perception, 2025.

[7] Z. Fu, T. Z. Zhao, and C. Finn. Mobile ALOHA: Learning bimanual mobile manipulation using low-cost whole-body teleoperation. In *8th Annual Conference on Robot Learning*, 2024.

[8] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023.

[9] T. He, J. Gao, W. Xiao, Y. Zhang, Z. Wang, J. Wang, Z. Luo, G. He, N. Sobanbabu, C. Pan, Z. Yi, G. Qu, K. Kitani, J. Hodgins, L. J. Fan, Y. Zhu, C. Liu, and G. Shi. Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills. *arXiv preprint arXiv:2502.01143*, 2025.

[10] L. Han, Q. Zhu, J. Sheng, C. Zhang, T. Li, Y. Zhang, H. Zhang, Y. Liu, C. Zhou, R. Zhao, J. Li, Y. Zhang, R. Wang, W. Chi, X. Li, Y. Zhu, L. Xiang, X. Teng, and Z. Zhang. Lifelike agility and play in quadrupedal robots using reinforcement learning and generative pre-trained models. *Nature Machine Intelligence*, 6(7):787–798, July 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00861-3.

[11] E. Vollenweider, M. Bjelonic, V. Klemm, N. Rudin, J. Lee, and M. Hutter. Advanced skills through multiple adversarial motion priors in reinforcement learning, 2022.

[12] C. Li, S. Blaes, P. Kolev, M. Vlastelica, J. Frey, and G. Martius. Versatile skill control via self-supervised adversarial imitation of unlabeled mixed motions. In *2023 IEEE international conference on robotics and automation (ICRA)*, pages 2944–2950. IEEE, 2023.

[13] C. Li, M. Vlastelica, S. Blaes, J. Frey, F. Grimminger, and G. Martius. Learning agile skills via adversarial imitation of rough partial demonstrations. In *Conference on Robot Learning*, pages 342–352. PMLR, 2023.

[14] J. Wu, G. Xin, C. Qi, and Y. Xue. Learning robust and agile legged locomotion using adversarial motion priors. *IEEE Robotics and Automation Letters*, 8(8):4975–4982, 2023. doi:10.1109/LRA.2023.3290509.

[15] L. Smith, J. C. Kew, T. Li, L. Luu, X. B. Peng, S. Ha, J. Tan, and S. Levine. Learning and adapting agile locomotion skills by transferring experience, 2023.

[16] X. B. Peng, E. Coumans, T. Zhang, T.-W. E. Lee, J. Tan, and S. Levine. Learning agile robotic locomotion skills by imitating animals. In *Robotics: Science and Systems*, 07 2020. doi:10.15607/RSS.2020.XVI.064.

[17] 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.*, 37(4):143:1–143:14, July 2018. ISSN 0730-0301. doi:10.1145/3197517.3201311.

[18] C. Li, E. Stanger-Jones, S. Heim, and S. Kim. Fld: Fourier latent dynamics for structured motion representation and learning. *arXiv preprint arXiv:2402.13820*, 2024.

[19] R. Watanabe, C. Li, and M. Hutter. Dfm: Deep fourier mimic for expressive dance motion learning, 2025.

[20] 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.*, 40(4), July 2021. doi:10.1145/3450626.3459670.

[21] 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. 2022 ieee. In *International Conference on Intelligent Robots and Systems (IROS)*, volume 2, 2022.

[22] X. B. Peng. *Acquiring Motor Skills Through Motion Imitation and Reinforcement Learning*. PhD thesis, EECS Department, University of California, Berkeley, Dec 2021.

[23] Z. Luo, J. Cao, J. Merel, A. Winkler, J. Huang, K. M. Kitani, and W. Xu. Universal humanoid motion representations for physics-based control. In *The Twelfth International Conference on Learning Representations*, 2024.

[24] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Trans. Graph.*, 41(4), July 2022.

[25] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. Anymal parkour: Learning agile navigation for quadrupedal robots. *Science Robotics*, 9(88):eadi7566, 2024. doi:10.1126/scirobotics.adi7566.

[26] L. Wellhausen and M. Hutter. Rough terrain navigation for legged robots using reachability planning and template learning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6914–6921, 2021. doi:10.1109/IROS51168.2021.9636358.

[27] F. Yang, C. Wang, C. Cadena, and M. Hutter. iplanner: Imperative path planning, 2023.

[28] C. Zhang, J. Jin, J. Frey, N. Rudin, M. Mattamala, C. Cadena, and M. Hutter. Resilient legged local navigation: Learning to traverse with compromised perception end-to-end. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 34–41, 2024. doi:10.1109/ICRA57147.2024.10611254.

[29] J. Lee, M. Bjelonic, A. Reske, L. Wellhausen, T. Miki, and M. Hutter. Learning robust autonomous navigation and locomotion for wheeled-legged robots. *Science Robotics*, 9(89): eadi9641, 2024. doi:10.1126/scirobotics.adi9641.

[30] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter. Advanced skills by learning locomotion and local navigation end-to-end, 2022.

[31] H. Zhang, S. Starke, T. Komura, and J. Saito. Mode-adaptive neural networks for quadruped motion control. *ACM Trans. Graph.*, 37(4), July 2018. ISSN 0730-0301. doi:10.1145/3197517.3201366.

[32] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5(47):eabc5986, 2020. doi:10.1126/scirobotics.abc5986.

[33] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl. Learning by cheating. In L. P. Kaelbling, D. Kragic, and K. Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 66–75. PMLR, 30 Oct–01 Nov 2020.

[34] 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.

[35] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017.

Appendix

7 Training Details

7.1 Details for FLD Model Training

As mentioned in Sec. 3.2, we train FLD model and use the FLD encoder to train low-level policy. The trained FLD encoder, through the propagation of latent dynamics, concurrently generates embeddings for global periodic parameters $\theta_t = (f_t, a_t, b_t)$, as well as for local phase states ϕ_t extracted from motion clips. The global parameters f_t, a_t, and b_t correspond to the frequency, amplitude, and offset of the latent trajectories. The parameterization leverages an autoencoder-like architecture to explicitly model the latent dynamics using a time-invariant frequency f_t and a fixed time step Δt with an autoencoder-like structure, as defined by:

$$\mathbf{z}_t = (\theta_t, \phi_t) = \mathbf{enc}(\mathbf{s}_t), \quad \hat{\mathbf{z}}_{t+i} = (\theta_t, \phi_t + i f_t \Delta t), \tag{4}$$

$$\hat{\mathbf{s}}_{t+i} = \mathbf{dec}(\hat{\mathbf{z}}_{t+i}), \quad L^{FLD} = \sum_{i}^{N} \text{MSE}(\mathbf{s}_{t+i}, \hat{\mathbf{s}}_{t+i}). \tag{5}$$

Here, $\mathbf{s}$ denotes the original motion sequence and $\mathbf{z}$ is latent representation, while $\mathbf{enc}(\cdot)$, and $\mathbf{dec}(\cdot)$ correspond to the encoding, and decoding operations, respectively. Our encoded state space comprises

the base's linear and angular velocities, the projected gravity vector in the robot base frame, and the joint positions. The overall loss L^{FLD} is computed as the reconstruction error between the reference and predicted state sequences over N consecutive segments, effectively capturing the motion's global periodic features. For further details on the training hyperparameters and network architecture of the FLD model, refer to Table 4 to Table 5 and consult original FLD paper.

7.2 Reward Setup for Low-Level and High-Level Policy Training

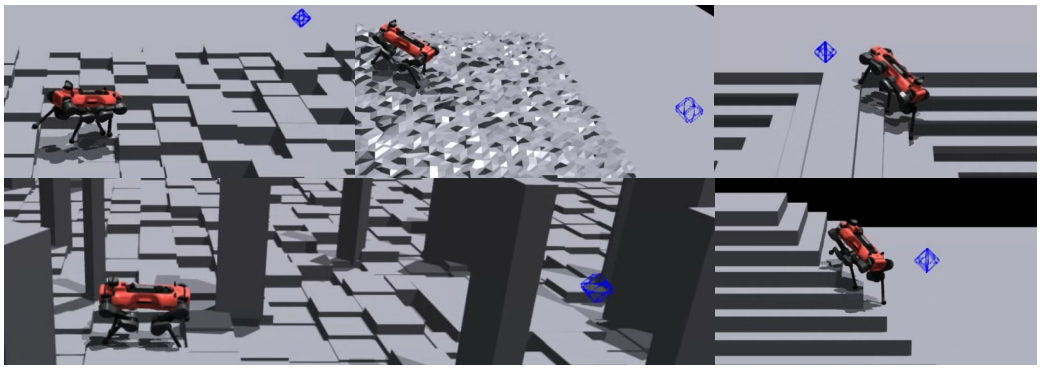

Figure 9: Overview of the training environment and terrain configuration for high-level policy. The setup includes diverse terrain types, boxes, stairs, rugged slopes, and high obstacles with the blue marker indicating the goal position.

The reward equations used for low-level policy and high-level policy are summarized in Table 1 to Table 3.

Table 1: Reward Equations for Low-Level Policy

Name	**Equation**
Linear Velocity Tracking	$2\exp\left(-\left\|\mathbf{v}_t^b - \mathbf{v}_t^{b,ref}\right\|^2\right)$
Angular Velocity Tracking	$0.8\exp\left(-0.8\left\|\mathbf{w}_t^b - \mathbf{w}_t^{b,ref}\right\|^2\right)$
Joint Position Tracking	$1.4\exp\left(-2\sum_{i=1}^{12}(q_{t,i} - q_{t,i}^{ref})^2\right)$
Projected Gravity Tracking	$0.8\exp\left(-3\left\|\mathbf{g}_t - \mathbf{g}_t^{ref}\right\|^2\right)$
Action Rate	$-0.005\sum_{i=1}^{12}(a_{t,i} - a_{t-1,i})^2$
Collision	$-\sum_{k\in\text{thigh, shanks}} c_k$
Torque Limits	$-0.2\sum_{i=1}^{12}\max(\tau_{t,i} - \tau_{lim}, 0)$
Torques	$-0.00002\sum_{i=1}^{12}\tau_{t,i}^2$
Joint Acceleration	$-0.00007\sum_{i=1}^{12}\ddot{q}_{t,i}^2$
Feet Acceleration	$-0.0001\sum_{i=1}^{4}\left\|\mathbf{v}_{t,i}^f - \mathbf{v}_{t-1,i}^f\right\|^2$
Contact Forces	$-0.005\sum_{i=1}^{4}{F_{t,i}^f}^2$

7.3 Training Setup and Hyperparameters

Hyperparameters for FLD model, low-level policy, high-level policy training are presented in Table 4 to Table 7. All simulations are performed in Isaac Gym [34]. Both the low-level policy and the high-level teacher policy are trained in parallel environments using the PPO algorithm [35] on a single NVIDIA RTX 4090. The low-level policy is trained exclusively on flat terrain, whereas the high-level policy is trained in the more complex environment illustrated in Fig. 9.

Table 2: Reward Equations for High-Level Policy

Name	Equation
Position Tracking	$15r_{reach}$
Heading Velocity	$5r_{vel}$
Joint Residual	$-0.1\sum_{i=1}^{12}(a_{t,i}^{res})^2$
Action Rate	$-0.005\sum_{i=1}^{12}(a_{t,i}^{res} - a_{t-1,i}^{res} + a_{t,i} - a_{t-1,i})^2$
Collision	$-\sum_{k\in\text{thigh, shanks}} c_k$
Stand Still	$-2.5\left\|\mathbf{v}_t^b\right\|^2 - \left\|\mathbf{w}_t^b\right\|^2$
Stand Pose	$-0.2\sum_{i=1}^{12}(q_{t,i} - q_i^*)^2 - 5(g_{x,t}^2 + g_{y,t}^2)$
Torque Limits	$-0.2\sum_{i=1}^{12}\max(\tau_{t,i} - \tau_{lim}, 0)$
Termination	-200
Torques	$-0.00002\sum_{i=1}^{12}\tau_{t,i}^2$
Joint Acceleration	$-0.00007\sum_{i=1}^{12}\ddot{q}_{t,i}^2$
Feet Acceleration	$-0.0001\sum_{i=1}^{4}\left\|\mathbf{v}_{t,i}^f - \mathbf{v}_{t-1,i}^f\right\|^2$
Contact Forces	$-0.005\sum_{i=1}^{4}{F_{t,i}^f}^2$

Table 3: Symbols for Table 1 and Table 2

Symbol	Description
ω_z^b	Yaw base velocity
c_k	1 if body k is in contact, 0 otherwise
$\mathbf{v}^b, \mathbf{v}_i^f$	linear velocity vector of base and foot i
$\mathbf{w}^b$	Angular velocity vector of base
q_i, q_i^*, q_i^{ref}	actual, default and reference position of joint i
$\ddot{q}_i$	acceleration of joint i
τ_i, τ_{lim}	torque and torque limit of joint i
F_i^f	Contact force of foot i
$\mathbf{g}, g_x, g_y$	Projected gravity vector, projected gravity along x and y axis in robot frame

Table 4: Training Hyperparameters for FLD

Configuration	Values
Step Time Seconds	0.02
Latent Channel	4
Propagation Horizon	30
Trajectory Segment Length	31
Propagation Decay	1.0
Learning Rate	0.0001
Weight Decay	0.0005
Number of Mini-Batches	20

Table 5: Network Architecture for FLD

Network	Layer	Output Size	Activation
Encoder	Conv1d	64×31	ELU
	Conv1d	64×31	ELU
	Conv1d	4×31	ELU
Phase Encoder	Linear	4×2	Atan2
Decoder	Conv1d	64×31	ELU
	Conv1d	64×31	ELU
	Conv1d	27×31	ELU

Table 6: Training Hyperparameters for PPO

Configuration	Values
Actor and Critic Network	MLP Hidden Layer Size (512, 256, 128)
Robot Number	4096
Step Number per Policy Update	48
Entropy Coefficient	0.002
Learning Rate	adaptive
GAE-lambda	0.95
Discount Factor	0.99
Coefficient of KL Divergence	0.01
Clip Ratio	0.2
Mini Batch Size	49152

Table 7: Training Hyperparameters for Student Policy

Configuration	Values
Truncate Step for TBPTT	15
Learning Rate	0.0005
Number of Learning Epochs	2

8 Additional Details for Experimental Results

8.1 Additional Results for Low-Level Policy

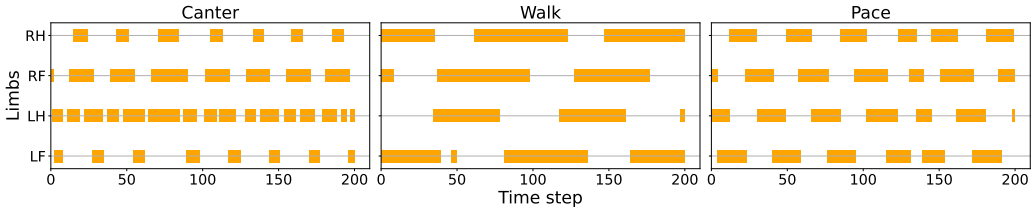

Figure 10: Footfall sequence of motion skills learned in low-level policy.

Figure 10 shows the footfall patterns of three flat-terrain skills in low-level policy under different latent embedding commands.

8.2 Evaluation on High-Level Student Policy in Simulation

We evaluate the high-level student policy with noisy observation in simulation and report its success rate for each terrain in Table 8. For each terrain type, we run 100 trials with the goal placed 5 meters from the robot's start in a random direction.

Table 8: Success Rate of High-Level Student Policy in Simulation

Terrain Type	Success Rate
0.25m Stairs (Up)	84/100
0.25m Stairs (Down)	85/100
24-deg Slope (Up)	90/100
24-deg Slope (Down)	95/100
Random Boxes	81/100
Random Boxes with High Obstacles	75/100
Flat ground with High Obstacles	90/100

8.3 Simulation Setup for Evaluation on Joint Residuals

As described in Sec. 4.2.2, we investigate how joint residuals vary across two terrain types, pyramid stairs and pyramid slopes, as the terrain difficulty progressively increases. The corresponding simulation setup is illustrated in Fig. 11.

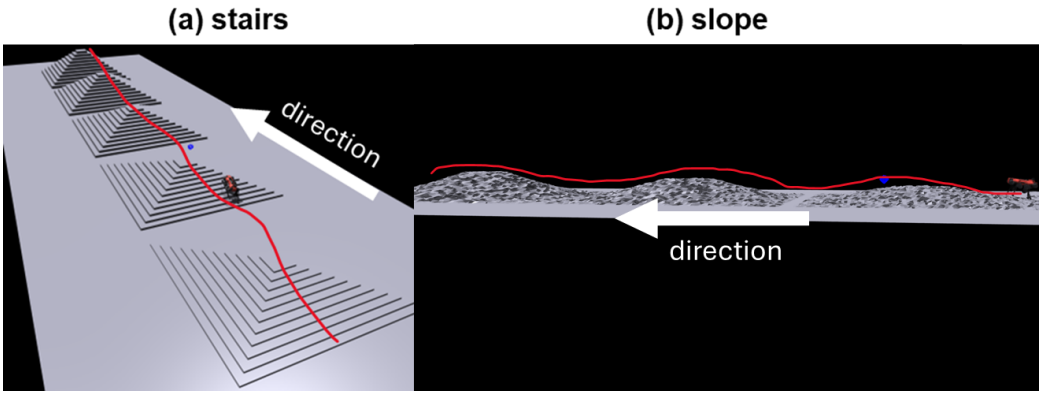

Figure 11: Overview of (a) pyramid stairs and (b) pyramid slope terrains with progressively increasing difficulty levels during inference. The blue marker indicates the goal position at the current frame which is moving from easier to more challenging terrain level, while the white arrow denotes the robot's walking direction. The red line traces an example trajectory of the robot's base position.

8.4 Additional Comparative Analysis on Energetic Efficiency

Building on the results in Sec. 4.2.4, we perform a comparative analysis of energetic efficiency of our models and the baseline under similar reward settings. In this experiment, each model continually walks forward by following a moving goal position. Figure 12 illustrates the Cost of Transport (CoT) on flat ground for the baseline and our two models, measured as the robot walks at varying speeds. CoT is calculated using the following equation.

$$CoT = \frac{\sum_i^{12} |\tau_{t,i}\dot{q}_{t,i}|}{mgv_b}, \tag{6}$$

where m denotes the total mass of the robot, $g(= 9.8\mathrm{m/s^2})$ the gravity constant, v_b the forward linear base velocity, $\dot{q}_{t,i}$ the velocity of joint i.

Although our framework does not explicitly optimize the energy consumption in the reward terms, our two models trained with motion priors achieve a lower CoT. This likely reflects the baseline model's insufficient constraints on vertical motion, whereas the incorporation of motion priors could naturally avoid those jumping gaits. Additionally, our comparison shows that removing residual regularization leads to a slight increase in CoT, probably due to the unconstrained residuals.

These findings imply that the animal-like motion priors can have the potential to steer training toward energy efficient gaits. However, further research is still needed to quantify this benefit and compare it against explicitly adding energy penalties to the reward functions.

8.5 Training High-Level Policies with Minimal Reward Terms

To further investigate how effectively our framework reduces the effort required for reward shaping, we trained an additional high-level policy using a minimal set of reward terms (Table 9), employing the same low-level motion priors. As illustrated in the supplementary videos, the new policy also successfully achieves perceptive locomotion and local navigation using animal-like gaits across varied terrains, despite the minimal reward configuration. Notably, unlike the model shown in the main paper, this simplified reward setup leads the policy to adopt a cantering gait rather than a walking gait. With the integration of learned joint residuals, the new high-level policy can still effectively traverse challenging terrains while maintaining the cantering gait. These results further confirm that

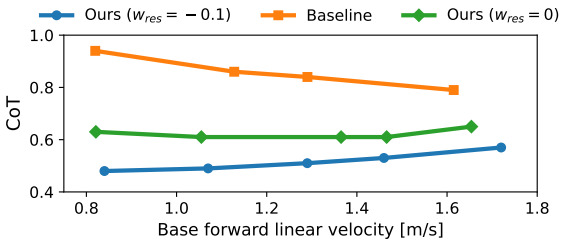

Figure 12: Energy efficiency comparison among three models: (i) the baseline RL model trained from scratch, (ii) our proposed framework trained with motion priors and a proper joint residual penalty ($w_{res} = -0.1$) (iii) a variant of our model without residual penalization ($w_{res} = 0$). The evaluation spans a wide range of forward velocities achieved via varying maximum velocity commands v_{cmd}. The CoT is averaged over a 20-second period of continuous forward motion.

Table 9: Reward Equations for High-Level Policy in Extended Experiments

Name	**Equation**
Position Tracking	$15r_{reach}$
Heading Velocity	$5r_{vel}$
Joint Residual	$-0.5\sum_{i=1}^{12}(a_{t,i}^{res})^2$
Collision	$-\sum_{k\in\text{thigh, shanks}} c_k$
Termination	-200

our framework can efficiently produce high-level policies with natural, animal-like locomotion and local navigation across diverse terrains, all while reducing reward complexity and tuning effort.