Multi-Loco: Unifying Multi-Embodiment Legged Locomotion via Reinforcement Learning Augmented Diffusion

Shunpeng Yang[*1], **Zhen Fu**[*1], **Zhefeng Cao**[1], **Junde Guo**[1],
Patrick Wensing[2], **Wei Zhang**[1,4], **Hua Chen**[3,4]
[1] Southern University of Science and Technology, China. [2] University of Notre Dame, United States
[3] Zhejiang University-University of Illinois Urbana-Champaign Institute, China
[4]LimX Dynamics, Shenzhen, China. * Equal contribution

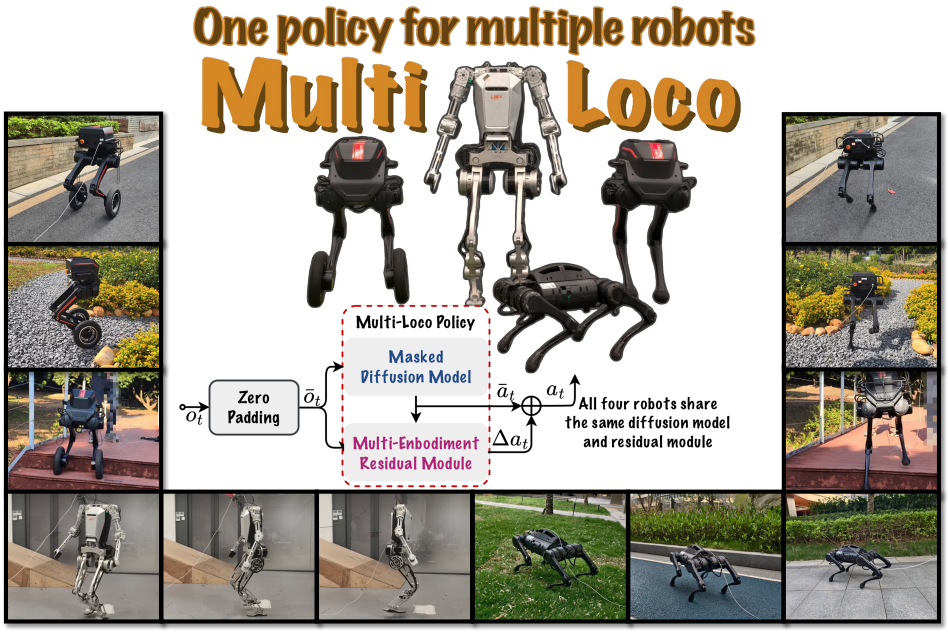


Figure 1: Deployment of the reinforcement learning augmented diffusion policy on four platforms (biped, wheeled biped, humanoid and quadruped). The experimental results demonstrate that the unified policy can effectively control the robots across various types of uneven terrain, including grass, slopes, stairs, and gravel paths. These results highlight the policy's robustness and exceptional control capabilities.

Abstract: Generalizing locomotion policies across diverse legged robots with varying morphologies is a key challenge due to differences in observation/action dimensions and system dynamics. In this work, we propose *Multi-Loco*, a novel unified framework combining a morphology-agnostic generative diffusion model with a lightweight residual policy optimized via reinforcement learning (RL). The diffusion model captures morphology-invariant locomotion patterns from diverse cross-embodiment datasets, improving generalization and robustness. The residual policy is shared across all embodiments and refines the actions generated by the diffusion model, enhancing task-aware performance and robustness for real-world deployment. We evaluated our method with a rich library of four legged robots in both simulation and real-world experiments. Compared to a standard RL framework with PPO, our approach - replacing the Gaussian policy with a diffusion model and residual term - achieves a 10.35% average return improvement, with gains up to 13.57% in wheeled-biped locomotion tasks. These results highlight the benefits of cross-embodiment data and composite generative architectures in learning robust, generalized locomotion skills. Project website: https://multi-loco.github.io

Keywords: Locomotion, Legged Robots, Multi-Embodiment, Diffusion Model, Reinforcement Learning

1 Introduction

Generalizing locomotion policies across legged robots with diverse embodiments is a fundamental challenge in robotics, but it offers a promising path toward scalable, efficient learning by enabling knowledge reuse and reducing platform-specific engineering. Although reinforcement learning (RL) has led to impressive progress in training locomotion controllers [1, 2, 3, 4], these policies are typically trained specifically for each robot morphology. Even robots with similar kinematic characteristics require separate training pipelines, limiting scalability and knowledge reuse. As a result, locomotion policies and datasets remain siloed, restricting unified learning across embodiments. While direct policy transfer to unseen robots remains an open problem, enabling policy generalization across diverse robots is an essential first step toward scalable multi-robot learning.

Recently, cross-embodiment learning has been attracting considerable research attention [5, 6, 7, 8]. These pioneering works have been conducted primarily in the domain of robotic manipulation tasks, aiming to develop generalizable policies that can handle variations in kinematics, sensing modalities, and control interfaces. However, transferring this success to legged locomotion presents unique challenges. Unlike manipulation, locomotion is deeply influenced by dynamical characteristics of the robots as well as their physical interaction with the environment, making it more difficult to abstract away embodiment-specific features. Moreover, reconciling the variations in observation and action dimensions across embodiments adds further complexity to the model design.

In this work, we propose *Multi-Loco*, a novel unified framework that addresses multi-embodiment locomotion learning through the integration of a generative diffusion model and a residual RL policy. Rather than relying on end-to-end transformer-based regression methods, we use a diffusion model to learn a morphology-invariant policy from diverse locomotion datasets, capturing generalizable patterns across robot embodiments. The output of the diffusion model is then refined by a lightweight residual policy trained with reinforcement learning, enhancing task-specific performance and adaptation, while remaining shared across all embodiments.

In summary, our key contributions are as follows:

- We propose a novel framework based on the integration of generative models with reinforcement learning to accommodate the design of unified locomotion policies across diverse legged platforms. Our approach captures shared locomotion patterns across various robots via leveraging the multi-modal capabilities of diffusion, and further bridges the sim2real gap with a shared residual policy trained via reinforcement learning.
- We leverage the multi-modal nature of diffusion models to conditionally model locomotion behaviors across diverse embodiments. By aligning heterogeneous observation and action spaces through zero padding and employing masked score matching during training, our framework enables a single generative policy to generalize across multiple robot platforms.
- We introduce a shared residual reinforcement learning policy that refines the diffusion model's action outputs to further bridge the sim2real gap. By specifying task-aware rewards and employing a multi-critic architecture, the single residual policy is capable of enhancing locomotion performance across the given embodiments, complementing the generative prior without requiring embodiment-specific tuning.

Related Work

Reinforcement learning for legged locomotion: RL has emerged as a powerful approach in the synthesis of robust and adaptable control policies in robotics, particularly for complex legged locomotion [1, 2, 3, 4, 9, 10, 11]. Hoeller et al. [12] demonstrated RL's capability to achieve agile parkour behaviors with quadrupedal robots, while Lee et al. [13] extended RL to wheeled-legged systems, tackling the combined challenges of locomotion and navigation. RL has also been applied to whole-body humanoid control. OmniH2O [14] combined teleoperation and RL to achieve dexterous full-body manipulation, and Hover [15] used neural controllers to unify locomotion and manipulation tasks. However, most of these methods require embodiment-specific training, with

limited generalizability across different platforms. The integration of RL with advanced generative models, such as diffusion models, to enhance cross-embodiment versatility remains underexplored.

Diffusion models and their applications in robotics: Diffusion models refine noisy samples through a learned denoising process, approximating complex, high-dimensional distributions. Song et al. [16] established a theoretical foundation connecting diffusion to stochastic differential equations (SDEs), enabling accelerated sampling via ODE solvers [17, 18] and further inspiring frameworks such as the Elucidated Diffusion Model (EDM) [19]. In robotics, diffusion models have shown promising applications for their ability to represent multi-modal behavior. Chi et al. [20] applied them to contact-rich manipulation tasks, while Huang et al. [21] introduced DiffuseLoco, one of the first applications of diffusion models to legged locomotion, demonstrating offline training and robust online control abilities. The potential of diffusion models for cross-embodiment locomotion learning remains to be further explored.

Cross-embodiment learning of legged robots: Cross-embodiment learning aims to develop control policies that generalize across robots with varying morphologies, actuation, and sensing. Early work in this field focused on morphology-specific adaptation, such as GenLoco [22], which trained quadruped controllers through procedural morphology randomization. However, such an approach was limited to robots with fixed degrees of freedom (DoFs). Subsequent investigations such as ManyQuadrupeds [23] generalizes to diverse quadruped morphologies by combining Central Pattern Generators (CPGs) with reinforcement learning, though their task-space control paradigm requires predefined robot-specific inverse kinematics. To address morphological variability, MorAL [24] introduced a morphology-aware network (MorphNet) that encodes physical information from proprioceptive observations. By conditioning policy learning on a compact morphology representation, MorAL improved robustness and generalization. Such an approach still relied on explicit morphology embeddings. Recently, URMA [25] employed transformers to unify control across quadrupeds, bipeds, and hexapods via morphology-agnostic encoders, demonstrating effective transfer with explicit joint descriptions for decoding. CrossFormer [8] further investigated generalist policies across legged platforms, yet their reliance on structured embodiment descriptors or per-robot modules may hinder scalability to new morphologies.

Table 1: Comparison of cross-embodiment learning models

Method	Morphology Agnostic	Independent of Joint Information	Cross Tasks	Generative Model
GenLoco [22]	×	✓	×	×
ManyQuadrupeds [23]	×	×	×	×
MorAL [24]	×	×	×	×
URMA [25]	✓	×	×	×
CrossFormer [8]	✓	×	✓	×
Multi-Loco (proposed)	✓	✓	×	✓

While these approaches have shown promising results in multi-embodiment learning, they often incorporate explicit morphology information - such as structural descriptors or per-robot observation encoders - to facilitate generalization. This design choice may present challenges for scaling to new robots or for learning in more morphology-agnostic settings. In this context, frameworks that aim to unify control across embodiments without relying on embodiment-specific inputs or architecture components remain relatively less explored.

2 Methodology

We propose *Multi-Loco*, a unified policy framework for multi-embodiment locomotion that combines generative modeling and reinforcement learning. Without relying on explicit morphology inputs, Multi-Loco captures shared locomotion patterns across diverse robots through a common policy structure. The framework consists of three components: (1) dimension alignment via zero-padding; (2) a diffusion model trained with masked denoising score matching; and (3) a residual reinforcement learning policy with multi-critic architecture. Details are provided below.

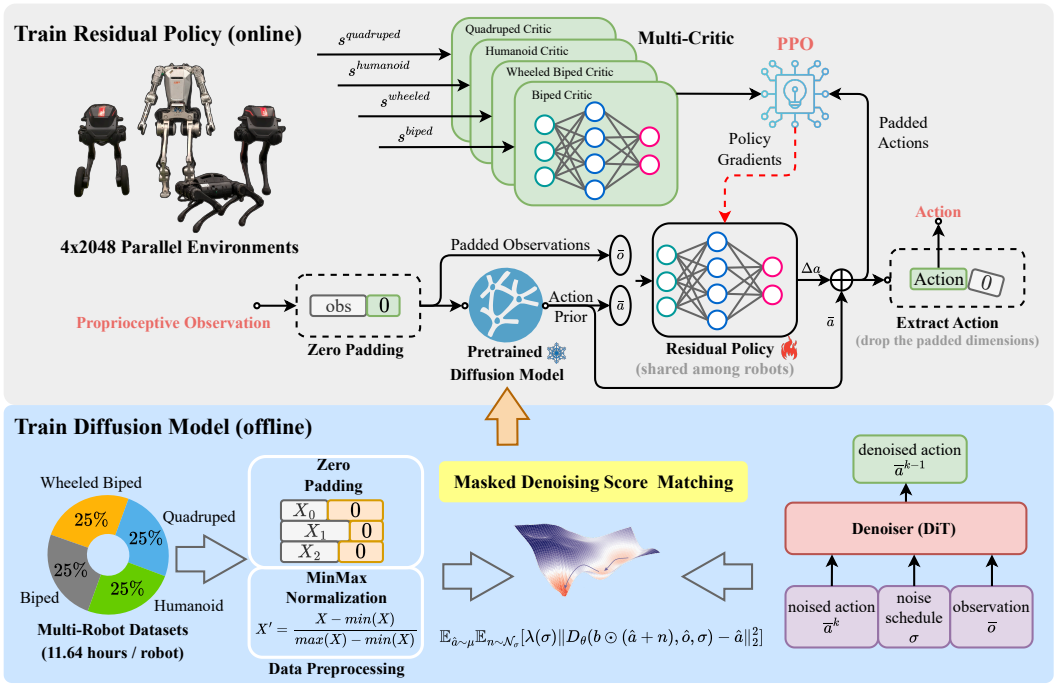


Figure 2: Overview of the *Multi-Loco* framework. Multi-robot datasets are preprocessed via zero-padding and normalization to align observation and action spaces across embodiments. A shared diffusion model is trained offline using masked denoising score matching. At inference time, the diffusion model generates action priors, which are refined by a residual policy trained via multi-critic PPO. Each critic specializes in one robot type, while the policy remains shared across all embodiments.

2.1 Dimension Alignment for Multi-Embodiment Observations and Actions

Multi-embodiment learning presents a fundamental challenge due to variations in observation and action spaces across different embodiments. To unify the policy space, we align all robot data into common observation and action spaces via zero-padding

$$\dim(\bar{\mathcal{O}}) = \max_{m \in \mathcal{M}} \dim(\mathcal{O}_m), \quad \dim(\bar{\mathcal{A}}) = \max_{m \in \mathcal{M}} \dim(\mathcal{A}_m)$$

where $\mathcal{M}$ denotes the set of robot embodiments, and $\bar{\mathcal{O}}$, $\bar{\mathcal{A}}$ are the unified observation and action spaces, respectively. For any robot m, its proprioceptive observation $\boldsymbol{o}_m \in \mathcal{O}_m$ and action $\boldsymbol{a}_m \in \mathcal{A}_m$ are mapped into $\bar{\boldsymbol{o}}_m \in \bar{\mathcal{O}}$ and $\bar{\boldsymbol{a}}_m \in \bar{\mathcal{A}}$ by padding zeros to the unused dimensions. In parallel, we define a binary mask $\boldsymbol{b} \in \{0,1\}^{\dim(\bar{\mathcal{A}})}$ that indicates which action dimensions are valid (1) or padded (0). For example, if a quadruped robot has a 12-dimensional action space and $\dim(\bar{\mathcal{A}}) = 20$, then its mask is $\boldsymbol{b} = [\underbrace{1,\ldots,1}_{12},\underbrace{0,\ldots,0}_{8}]$.

To improve training stability across heterogeneous embodiments, we further apply MinMax normalization to all observation and action dimensions. This normalization ensures consistent numerical ranges across robots, facilitating better convergence during diffusion model training and allowing shared model parameters to generalize effectively across varied scales.

This preprocessing standardizes observation-action formats while preserving the structural differences between embodiments. The resulting unified dataset forms the input for training a generative policy model, as described in the next section.

2.2 Multi-Embodiment Locomotion Unification via Diffusion Model

Learning robust policies across embodiments requires addressing multi-modality in robot dynamics and control strategies. To address this challenge, we propose employing diffusion models to directly parameterize the robot's action distribution $\mu(\bar{\boldsymbol{a}}|\bar{\boldsymbol{o}})$, leveraging their demonstrated advantages in

sample quality and robustness over alternative approaches such as VAEs and flow models, defined over the robot's action space $\bar{\boldsymbol{a}} \in \bar{\mathcal{A}}$ and proprioceptive observation space $\bar{\boldsymbol{o}} \in \bar{\mathcal{O}}$.

To ensure real-time inference, we adopt the EDM with a lightweight Diffusion Transformer (DiT) [26] backbone. EDM approximates the Stein score function $\nabla_{\bar{\boldsymbol{a}}} \log \mu_\sigma(\bar{\boldsymbol{a}}|\bar{\boldsymbol{o}})$ and reconstructs samples by solving ODEs. The denoising objective is

$$\mathbb{E}_{\bar{\boldsymbol{a}}\sim\mu}\mathbb{E}_{\boldsymbol{n}\sim\mathcal{N}_\sigma}[\lambda(\sigma)\|D_\theta(\bar{\boldsymbol{a}}+\boldsymbol{n},\bar{\boldsymbol{o}},\sigma)-\bar{\boldsymbol{a}}\|_2^2] \tag{1}$$

where D_θ is the denoiser network. To handle padded dimensions, we introduce masked denoising score matching to ensure that the model focuses only on valid data entries. Here, $\odot$ denotes element-wise multiplication (Hadamard product).

$$\mathbb{E}_{\bar{\boldsymbol{a}}\sim\mu}\mathbb{E}_{\boldsymbol{n}\sim\mathcal{N}_\sigma}[\lambda(\sigma)\|\boldsymbol{b}\odot\big(D_\theta(\boldsymbol{b}\odot(\bar{\boldsymbol{a}}+\boldsymbol{n}),\bar{\boldsymbol{o}},\sigma)-\bar{\boldsymbol{a}}\big)\|_2^2] \tag{2}$$

2.3 Enhancing Diffusion with Residual Policy and Multi-Critic RL

Although diffusion models provide strong priors, they may not fully capture fine-grained dynamics or task-specific variations. To address this, we introduce a residual policy shared across all robots

$$\boldsymbol{a} = \bar{\boldsymbol{a}}_{\text{prior}} + \Delta\boldsymbol{a} \tag{3}$$

where $\bar{\boldsymbol{a}}_{\text{prior}}$ is sampled from the diffusion model, and $\Delta\boldsymbol{a}$ is predicted by a residual policy $\pi_\theta(\Delta\boldsymbol{a}|\bar{\boldsymbol{o}},\bar{\boldsymbol{a}}_{\text{prior}})$ trained via PPO.

To enable generalization across robot embodiments, we train a single residual policy shared by all robots, while adopting a multi-critic PPO framework [27, 28, 29, 30] to resolve optimization conflicts. Specifically, we instantiate a separate critic V_ϕ^k for each robot type $k = 1,\ldots,K$, each receiving its corresponding privileged state $\boldsymbol{s}_t^k$ as input. During training, the shared actor π_θ is optimized using the following PPO loss:

$$L^{\text{PPO}}(\theta) = \sum_{k=1}^{K}\mathbb{E}_t\left[\min\left(r_t^k(\theta)\,\bar{A}_t^k,\ \text{clip}\big(r_t^k(\theta),\,1-\epsilon,\,1+\epsilon\big)\,\bar{A}_t^k\right)\right] \tag{4}$$

where $r_t^k(\theta) = \frac{\pi_\theta(\Delta\boldsymbol{a}_t^k|\bar{\boldsymbol{o}}^k,\bar{\boldsymbol{a}}_{\text{prior}}^k)}{\pi_{\theta_{\text{old}}}(\Delta\boldsymbol{a}_t^k|\bar{\boldsymbol{o}}^k,\bar{\boldsymbol{a}}_{\text{prior}}^k)}$ and $\bar{A}_t^k$ denotes the normalized advantage from the k-th critic.

Each robot's reward function includes (i) task-specific components (e.g., velocity tracking, stability), (ii) regularization terms (e.g., torque penalties), and (iii) a residual penalty $r_d(\Delta\boldsymbol{a}_t)$ that encourages the residual to remain close to the generative prior, reducing overcorrection. The complete reward formulation is detailed in Appendix B.

During training, we run multiple robots in parallel environments and alternate between diffusion sampling and residual optimization. Only the residual policy is updated online, while the diffusion model remains frozen. This design allows the residual to specialize to task-level feedback while leveraging the shared generative prior.

3 Evaluation

We conducted comprehensive experiments to validate two core hypotheses:

- ***Cross-embodiment generalization:*** Our unified policy framework outperforms robot-specific baselines in controlling diverse morphologies under identical RL hyperparameters.
- ***Zero-shot sim2real transfer:*** The proposed residual policy effectively bridges the sim2real gap for diffusion-based controllers.

Our ablation and real-world deployment tests address three critical research questions:

- **Cross-embodiment knowledge emergence:** Does cross-robot training on heterogeneous embodiments facilitate the emergence of shared locomotion skills that remain unattainable through single-embodiment learning paradigms?

- **Performance superiority:** Can our cross-robot policy outperform morphology-specific reinforcement learning baselines in locomotion tasks when trained under identical parameter constraints and environmental conditions?
- **Sim2real transfer:** Does the residual policy effectively enable zero-shot transfer of the multi-embodiment diffusion model to physical hardware?

3.1 Evaluation Setup

Our evaluation framework encompasses four distinct robots to systematically validate cross-morphology generalization capabilities: **(1) Point-foot biped** - Minimalist design with underactuated dynamics, **(2) Wheel-actuated biped** - Hybrid locomotion combining leg and wheel modalities, **(3) Full-scale humanoid** - High-DoF system with complex whole-body coordination, and **(4) Quadruped** - Dynamic quadrupedal locomotion with multiple contact points.

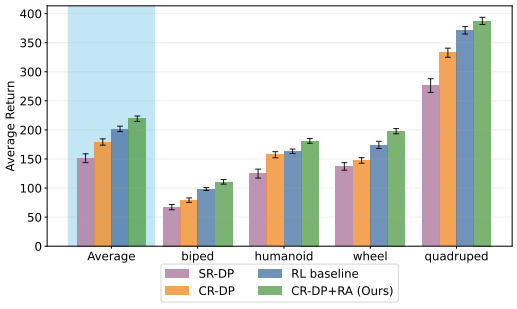
The multi-morphological platforms enable a comprehensive evaluation of our unified control framework through both simulation and real-world experiments. These tests demonstrate the framework's robustness and adaptability in managing embodied variations across different robotic morphologies. The subsequent sections detail our methodology and experimental results using these platforms.

3.2 Ablation Study

To demonstrate the significance of each module within the unified framework, we have designed the following ablation studies:

- **RL Baseline**: Morphology-specific policies trained via PPO, following implementation practices from `humanoid_gym` [1] and `legged_gym` [31].
- **Single-Robot Diffusion Policy (SR-DP)**: EDM trained on individual robot demonstrations.
- **Cross-Robot Diffusion Policy (CR-DP)**: Our EDM variant with dynamic action masking, trained on aggregated multi-robot datasets.
- **CR-DP with Residual Adaptation (CR-DP+RA)**: Our full approach integrating masked diffusion pre-training on multi-robot datasets with subsequent PPO-based residual policy adaptation through shared network parameters.

To ensure an equitable comparison across four methodologies, we maintained identical RL parameters when training both CR-DP+RA and RL baseline. Our evaluation protocol employs policies trained via baseline RL to generate rollout data for diffusion policy (DP) training, enabling direct comparison of average returns and auxiliary metrics within a unified environmental setup.



(a) Average Return of 4 Robots

(b) Humanoid Locomotion Experiments

(c) Point-Foot Biped Downstairs

Figure 3: (a) Comparative performance analysis of four robot morphologies (biped, humanoid, wheeled, quadruped) in terrain negotiation tasks. CR-DP+RA achieves 10.35% average improvement over RL baseline (12.49% biped, 13.57% wheeled-biped, 4.38% quadruped, 10.97% humanoid), while CR-DP surpasses SR-DP by 17.96% (17.81% biped, 7.52% wheeled-biped, 20.47% quadruped, 26.02% humanoid). (b) Humanoid locomotion demonstrates zero-shot sim2real transfer. (c) Point-foot biped successfully descends stairs

Our evaluation metrics are designed as follows:

- **(AR)** Average Return: Represents the cumulative RL return obtained by the agent during its survival period, up to a maximum of 20 seconds. It reflects the overall performance of the agent, incorporating both task success and control efficiency.
- **(MEL)** Mean Episode Length: Represents the average survival duration of the agent in each episode, with a maximum of 20 seconds. A longer episode length indicates better stability.
- **(LVT/AVT)** Mean Episode Linear/Angular Velocity Tracking Reward: Represents the reward associated with tracking a desired linear/angular velocity.

Table 2: Cross-Morphology Performance Comparison of Ablation Studies

Method	**Point-Foot**			**Wheeled**		
	MEL↑	LVT↑	AVT↑	MEL↑	LVT↑	AVT↑
Baseline-point-foot	$\mathbf{18.71 \pm 0.37}$	1.39 ± 0.25	0.94 ± 0.17	-	-	-
SR-DP-point-foot	14.68 ± 0.66	0.93 ± 0.37	0.71 ± 0.26	-	-	-
Baseline-wheeled	-	-	-	18.29 ± 0.37	2.48 ± 0.58	1.28 ± 0.30
SR-DP-wheeled	-	-	-	18.63 ± 0.33	2.37 ± 0.55	1.26 ± 0.28
CR-DP (Ours)	17.28 ± 0.56	1.15 ± 0.31	0.87 ± 0.21	$\mathbf{18.97 \pm 0.35}$	2.54 ± 0.50	1.35 ± 0.26
CR-DP+RA (Ours)	18.13 ± 0.46	$\mathbf{1.42 \pm 0.29}$	$\mathbf{0.97 \pm 0.20}$	18.86 ± 0.33	$\mathbf{2.72 \pm 0.53}$	$\mathbf{1.36 \pm 0.28}$
Method	**Humanoid**			**Quadruped**		
	MEL↑	LVT↑	AVT↑	MEL↑	LVT↑	AVT↑
Baseline-humanoid	18.88 ± 0.43	1.57 ± 0.27	1.23 ± 0.20	-	-	-
SR-DP-humanoid	13.84 ± 0.78	1.04 ± 0.42	0.93 ± 0.34	-	-	-
Baseline-quadruped	-	-	-	$\mathbf{19.79 \pm 0.17}$	$\mathbf{5.45 \pm 0.37}$	$\mathbf{4.34 \pm 0.30}$
SR-DP-quadruped	-	-	-	16.90 ± 0.62	4.51 ± 1.25	3.32 ± 0.95
CR-DP (Ours)	17.13 ± 0.54	1.33 ± 0.36	1.17 ± 0.29	19.31 ± 0.29	5.29 ± 0.57	4.19 ± 0.47
CR-DP+RA (Ours)	$\mathbf{19.11 \pm 0.44}$	$\mathbf{1.58 \pm 0.28}$	$\mathbf{1.35 \pm 0.23}$	19.60 ± 0.19	5.42 ± 0.50	4.32 ± 0.41

3.2.1 Emergence of Shared Locomotion Skills via Cross-Embodiment Learning

Emergence of shared locomotion skills via cross-embodiment learning refers to the phenomenon where robotic agents with distinct morphologies learn fundamental movement patterns through shared learning frameworks.

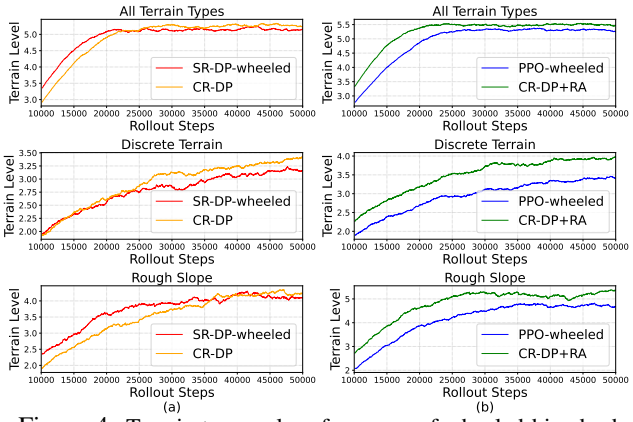


Figure 4: Terrain traversal performance of wheeled-biped robots under different training setups. (a) Performance of the diffusion model trained using offline datasets only (SR-DP-Wheeled vs. CR-DP) on terrains seen during training. (b) Comparison between PPO-trained baseline and our diffusion+residual policy (CR-DP+RA), which incorporates online reinforcement learning. CR-DP shows improved adaptability over SR-DP-Wheeled on rougher terrains, while CR-DP+RA further enhances robustness and outperforms PPO in terms of stability and obstacle traversal.

As evidenced in Figure 4, our proposed CR-DP achieves an overall terrain traversal improvement of 2.67% over SR-DP for wheeled-bipeds, with particularly surges in challenging terrains: 6.10% improvement over discrete obstacles and 3.87% on rough slopes, demonstrating its adaptability to complex environments.

Notably, under identical parameter configurations, the CR-DP with Residual Adaptation variant demonstrates measurable performance gains over the RL Baseline in wheeled-biped scenarios, achieving a 15.13% improvement on discrete terrain, 11.57% enhancement on rough slopes, and an overall terrain navigation improvement of 3.46%. This is particularly remarkable given that the original wheeled-biped robot dataset and RL training parameters failed to enable leg-lifting locomotion - which fundamentally restricted mobility in discontinuous terrains such as stepped surfaces and stair-like obstacles. Our experiments demonstrate that the cross-robot

DP component effectively transfers terrain negotiation expertise from heterogeneous embodiments to the wheeled-biped platform, enabling stable traversal of challenging terrains.

Figure 3 reveals that CR-DP achieves comprehensively higher average returns than SR-DP, while CR-DP with Residual Adaptation substantially exceeds RL Baseline performance, which achieves a 10.35% improvement on average, with gains up to 13.57% in wheeled-biped locomotion tasks. These results strongly suggest that the cross-robot component successfully captures shared locomotion skills. An interesting observation emerges in quadruped scenarios: CR-DP with Residual Adaptation matches RL Baseline performance in MEL and LVT/AVT Reward metrics. We hypothesize this reflects asymmetric knowledge transfer - quadruped locomotion experiences provide exemplary guidance for other morphologies, while motion patterns from other embodiments offer limited benefit to quadruped agents.

3.2.2 Multi-Loco Policy Outperforms Morphology-Specific RL Baseline

As quantitatively demonstrated in Figure 3 and Table 2, our CR-DP with Residual Adaptation framework achieves consistent performance advantages over the RL Baseline across all evaluation metrics and four robotic embodiments. The supervised learning foundation of CR-DP reaches 87.78% of the baseline RL's maximum average return potential, consistent with known limitations of behavioral cloning paradigms [32]. Through integration with Residual Adaptation - implemented using identical RL hyperparameters - the hybrid architecture demonstrates significant performance improvements, attaining 113.57% of baseline effectiveness. This performance inversion reveals the critical role of our hybrid learning framework: while the diffusion model component distills transferable motion skills from heterogeneous robot experiences, the residual RL module dynamically adapts these skills to embodiment-specific dynamics through online RL.

3.2.3 Zero-Shot sim2real Transfer: Experiments Evaluation for the Robust Locomotion

We implemented the unified locomotion policy across all four robots (embodiment details in Fig.2), with real-time control executed on an Intel i9-13900HX CPU at 50Hz refresh rates using EtherCAT communication.

As demonstrated in Fig.1 and Fig.3(b)(c), our Multi-Loco policy enables robust locomotion capabilities across challenging environments. The framework facilitates smooth embodiment transitions and stable gait generation in diverse terrain geometries ranging from inclined surfaces to vegetated landscapes and irregular substrates. This operational consistency across heterogeneous environmental conditions confirms the policy's capacity to maintain dynamic stability through adapting its motor patterns, highlighting its practical viability for deployment in unstructured real-world settings.

4 Conclusions

In this work, we introduced Multi-Loco, a unified framework for multi-embodiment legged locomotion that integrates generative diffusion models with reinforcement learning. Our approach successfully addresses the challenge of generalizing control policies across diverse robot embodiments, enabling seamless adaptation to varying hardware configurations.

Experimental results across simulation and hardware platforms demonstrate the framework's ability to generalize effectively, highlighting its potential for real-world applications in dynamic and unpredictable environments. By leveraging masked diffusion model, Multi-Loco gracefully handles differences in observation and action spaces, ensuring scalability across diverse robotic systems.

In conclusion, Multi-Loco pioneers a novel pathway for tackling the challenges of multi-embodiment unified control policy, delivering a scalable and flexible framework that enables robust and adaptive locomotion across diverse robotic embodiments.

5 Limitations and Future Works

The proposed Multi-Loco framework demonstrates strong performance in unifying locomotion policies for diverse embodiments, including point-foot biped, wheeled biped, quadruped and humanoid, marking a foundational step toward general-purpose control policy synthesis for multi-embodiment robotic systems. However, several limitations remain, which we discuss below alongside potential future directions.

A key limitation of the current framework is its reliance on observation-action paired datasets for training. This restricts its ability to leverage widely available motion capture data (e.g., human or animal locomotion), which often lack explicit action labels but encode rich motor skills. Future work will focus on action-free training paradigms to integrate such motion data, enabling the framework to exploit larger and more diverse datasets.

While Multi-Loco successfully unifies policies for predefined embodiments, direct synthesis of locomotion policies for entirely new morphologies remains an open challenge. Future efforts will investigate how shared representations of locomotion principles, learned from existing embodiments, can be extended to novel morphologies through zero-shot adaptation or few-shot fine-tuning, reducing the need for extensive retraining.

By addressing these challenges, we aim to advance Multi-Loco into a universal locomotion synthesis framework capable of generalizing across embodiments, tasks, and real-world conditions.

References

[1] X. Gu, Y.-J. Wang, X. Zhu, C. Shi, Y. Guo, Y. Liu, and J. Chen. Advancing humanoid locomotion: Mastering challenging terrains with denoising world model learning. *arXiv preprint arXiv:2408.14472*, 2024.

[2] 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.

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

[4] 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.

[5] J. Yang, D. Sadigh, and C. Finn. Polybot: Training one policy across robots while embracing variability. *arXiv preprint arXiv:2307.03719*, 2023.

[6] M. Xu, Z. Xu, C. Chi, M. Veloso, and S. Song. Xskill: Cross embodiment skill discovery. In *Conference on Robot Learning*, pages 3536–3555. PMLR, 2023.

[7] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[8] R. Doshi, H. Walke, O. Mees, S. Dasari, and S. Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. *arXiv preprint arXiv:2408.11812*, 2024.

[9] H. Wang, H. Luo, W. Zhang, and H. Chen. Cts: Concurrent teacher-student reinforcement learning for legged locomotion. *IEEE Robotics and Automation Letters*, 2024.

[10] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme parkour with legged robots. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11443–11450. IEEE, 2024.

[11] K. Jiang, Z. Fu, J. Guo, W. Zhang, and H. Chen. Learning whole-body loco-manipulation for omni-directional task space pose tracking with a wheeled-quadrupedal-manipulator. *IEEE Robotics and Automation Letters*, 2024.

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

[13] 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.

[14] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

[15] T. He, W. Xiao, T. Lin, Z. Luo, Z. Xu, Z. Jiang, J. Kautz, C. Liu, G. Shi, X. Wang, et al. Hover: Versatile neural whole-body controller for humanoid robots. *arXiv preprint arXiv:2410.21229*, 2024.

[16] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[17] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[18] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

[19] T. Karras, M. Aittala, T. Aila, and S. Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

[20] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[21] X. Huang, Y. Chi, R. Wang, Z. Li, X. B. Peng, S. Shao, B. Nikolic, and K. Sreenath. Diffuseloco: Real-time legged locomotion control with diffusion from offline datasets. *arXiv preprint arXiv:2404.19264*, 2024.

[22] G. Feng, H. Zhang, Z. Li, X. B. Peng, B. Basireddy, L. Yue, Z. Song, L. Yang, Y. Liu, K. Sreenath, et al. Genloco: Generalized locomotion controllers for quadrupedal robots. In *Conference on Robot Learning*, pages 1893–1903. PMLR, 2023.

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

[24] Z. Luo, Y. Dong, X. Li, R. Huang, Z. Shu, E. Xiao, and P. Lu. Moral: Learning morphologically adaptive locomotion controller for quadrupedal robots on challenging terrains. *IEEE Robotics and Automation Letters*, 2024.

[25] N. Bohlinger, G. Czechmanowski, M. Krupka, P. Kicki, K. Walas, J. Peters, and D. Tateo. One policy to run them all: an end-to-end learning approach to multi-embodiment locomotion. *arXiv preprint arXiv:2409.06366*, 2024.

[26] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[27] S. Mysore, G. Cheng, Y. Zhao, K. Saenko, and M. Wu. Multi-critic actor learning: Teaching rl policies to act with style. In *International Conference on Learning Representations*, 2022.

[28] P. Xu, X. Shang, V. Zordan, and I. Karamouzas. Composite motion learning with task control. *ACM Transactions on Graphics (TOG)*, 42(4):1–16, 2023.

[29] T. Huang, J. Ren, H. Wang, Z. Wang, Q. Ben, M. Wen, X. Chen, J. Li, and J. Pang. Learning humanoid standing-up control across diverse postures. *arXiv preprint arXiv:2502.08378*, 2025.

[30] H. Wang, Z. Wang, J. Ren, Q. Ben, T. Huang, W. Zhang, and J. Pang. Beamdojo: Learning agile humanoid locomotion on sparse footholds. *arXiv preprint arXiv:2502.10363*, 2025.

[31] 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.

[32] S. Ross and D. Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668. JMLR Workshop and Conference Proceedings, 2010.

[33] D. Picard. Torch. manual_seed (3407) is all you need: On the influence of random seeds in deep learning architectures for computer vision. *arXiv preprint arXiv:2109.08203*, 2021.

[34] Z. Dong, Y. Yuan, J. Hao, F. Ni, Y. Ma, P. Li, and Y. Zheng. Cleandiffuser: An easy-to-use modularized library for diffusion models in decision making. *Advances in Neural Information Processing Systems*, 37:86899–86926, 2024.

Appendix

Experiment Videos

We conducted comprehensive real-world evaluations of our framework across four distinct legged robotic platforms. For detailed empirical validation, we encourage readers to view the supplementary video. As demonstrated in the experimental recordings, our framework demonstrates the capability of a unified control policy to govern four morphologically diverse legged robots with different actuator configurations and mass distributions. This cross-platform adaptability is achieved through our novel methodology, which enables robust policy generalization while maintaining dynamic locomotion performance.

A Details of Diffusion Model Training and Inference

DDPM, as one kind of diffusion models, has achieved remarkable results in tasks such as robotic manipulation and locomotion for legged robots due to its ability to represent multimodality and complex distributions. Due to the high number of denoising steps required by DDPM, the inference time can be quite long. For humanoid robots, the diffusion policy must operate at a frequency of at least 50 Hz, ideally reaching 100 Hz. Therefore, it is necessary to either accelerate its performance or replace it with alternatives such as DDIM, EDM, or consistency models. In this context, we have chosen the EDM model to ensure high sample quality while minimizing the inference time to enable high-frequency feedback control.

EDM (Elucidated Diffusion Model) [19] is based on the reverse-time ODE of variance-exploding SMLD [16]. In order to elucidate the design space of diffusion model, EDM proposes a more general form while considering time-dependent scaling and introduces preconditioning to cancel the effects caused by increasing noise variance in denoising score matching. EDM reparametrized the denoiser $D_\theta(\hat{\boldsymbol{x}}, \sigma)$ as the following form

$$D_\theta(\hat{\boldsymbol{x}}, \sigma) = c_{\text{skip}}(\sigma)\hat{\boldsymbol{x}} + c_{\text{out}}(\sigma)F_\theta(c_{\text{in}}(\sigma)\hat{\boldsymbol{x}}, c_{\text{noise}}(\sigma)). \tag{5}$$

and the corresponding objective of denoising score matching is changed to

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{n}}\left[\frac{c_{\text{out}}(\sigma)^2}{\sigma}\|F_\theta(c_{\text{in}}(\sigma)(\boldsymbol{x}+\boldsymbol{n}), c_{\text{noise}}(\sigma)) - \boldsymbol{x}_d\|_2^2\right] \tag{6}$$

where $\boldsymbol{x}_d = (\boldsymbol{x} - c_{\text{skip}}(\sigma)(\boldsymbol{x}+\boldsymbol{n}))/c_{\text{out}}$. With the requirements of unit variance of input and output while minimizing c_{out}, the value of parameters are chosen as below:

$$c_{\text{noise}}(\sigma) = \ln(\sigma)/4, \tag{7a}$$

$$c_{\text{in}}(\sigma) = 1/\sqrt{\sigma_{\boldsymbol{x}}^2 + \sigma^2}, \tag{7b}$$

$$c_{\text{skip}}(\sigma) = \sigma_{\boldsymbol{x}}^2/(\sigma^2 + \sigma_{\boldsymbol{x}}^2), \tag{7c}$$

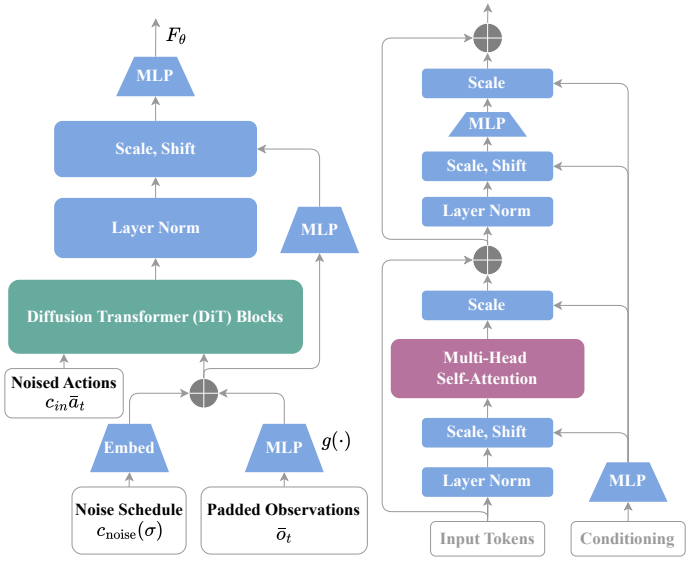
$$c_{\text{out}}(\sigma) = \sigma \cdot \sigma_{\boldsymbol{x}}/(\sigma^2 + \sigma_{\boldsymbol{x}}^2), \tag{7d}$$

where $\sigma_{\boldsymbol{x}}$ is the variance of the data. [19] suggests that the optimal choice for this function is $\sigma(t) = t$, an approach we also adopt in our work. As a result, EDM solves the following probability flow ODE

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \frac{\boldsymbol{x} - D_\theta(\boldsymbol{x}, \sigma)}{t} \tag{8}$$

for sampling and starting from $\boldsymbol{x}(T) \sim \mathcal{N}(0, T^2\boldsymbol{I})$ and stopping at $\boldsymbol{x}(0)$. In practice, we use Euler method to solve this ODE for sampling.

In terms of network architecture, diffusion models commonly utilize U-Net and Transformer-based structures. In our scenario, we chose DiT (Denoising Transformer) model [26] as backbone to fit $F_\theta(\cdot)$. The padded observations $\bar{\boldsymbol{o}}_t$ as condition variable was embed into a latent space by a multilayer perceptron (MLP) denoted by $g(\cdot)$. To incorporate $\bar{\boldsymbol{o}}_t$ as condition variable, $D_\theta(\hat{\boldsymbol{x}}, \sigma)$ is rewriten as $D_\theta(\hat{\boldsymbol{a}}_t, \bar{\boldsymbol{o}}_t, \sigma)$ and $F_\theta(c_{\text{in}}\hat{\boldsymbol{x}}, c_{\text{noise}}(\sigma))$ is changed to $F_\theta(c_{\text{in}}\hat{\boldsymbol{a}}_t, g(\bar{\boldsymbol{o}}_t), c_{\text{noise}}(\sigma))$. The details of the network structure can be found in Fig. 5.



(a) DiT Network (b) DiT Block

Figure 5: Neural network structure of DiT which is used to fit F_θ.

Parameter	Value / Description
Data Samples	
Prediction Horizon	4
Observation Horizon	10
Maximum Samples per Robot	2048000
Batch Size	8192
Diffusion Network (DiT)	
Input Dimension	20
Embedding Dimension (emb_dim)	128
Model Dimension (d_model)	256
Number of Attention Heads (n_heads)	8
Network Depth (depth)	3
Timestep Embedding Type	Fourier
MLP Condition Network $g(\cdot)$	
Input Dimension	683
Output Dimension	emb_dim
Hidden Dimensions (hidden_dim)	[512, 256]
Activation Function	ELU
Training Parameters	
Learning Rate	3e-4
Optimizer	Adam
Learning Rate Scheduler	Cosine Annealing
EMA Rate	0.999
Number of Epochs	500
Seed	3407 (inspired by [33])

Table 3: Configurations for Unified Diffusion Model

Algorithm 1 Masked EDM Training and Inference

Initialize:
- Networks: DiT F_θ and Condition MLP g_ϕ
- EDM params: $\sigma_{\text{data}} = 0.5$, $\sigma_{\min} = 0.002$, $\sigma_{\max} = 80.0$, $\rho = 7.0$
- Noise sampling: $P_{\text{mean}} = -1.2$, $P_{\text{std}} = 1.2$
- Training: batch size $= 512$, learning rate $= 3 \times 10^{-4}$, EMA rate $= 0.999$

procedure DATAPREPROCESSING
  Extract trajectory segments (observations $\boldsymbol{o}$, actions $\boldsymbol{a}$)
  Apply zero-padding to handle variable-length sequences
  Compute quantile-based MinMax normalization:
  - Calculate 5% and 95% quantiles for each feature dimension
  - Normalize to [-1,1] range: $\boldsymbol{x}_{\text{norm}} = 2 \cdot \frac{\boldsymbol{x} - \boldsymbol{x}_{\min}}{\boldsymbol{x}_{\max} - \boldsymbol{x}_{\min}} - 1$
  Create condition vector by concatenating normalized obs and commands
  return Normalized dataset, normalization statistics
end procedure
procedure TRAIN(Dataset $\mathcal{D}$, Epochs $= 400$K)
  for each epoch **do**
    for batch $(\boldsymbol{o}, \boldsymbol{a}_0, \boldsymbol{b}) \sim \mathcal{D}$ **do**
      Sample noise $\sigma \sim e^{\mathcal{N}(-1.2, 1.2^2)}$, $\epsilon \sim \mathcal{N}(0, I)$
      Compute noisy actions $\boldsymbol{a}_t = \boldsymbol{a}_0 + \sigma \cdot \epsilon$
      Compute EDM preconditioning coefficients:
      $c_{\text{skip}} = \frac{0.5^2}{0.5^2 + \sigma^2}$
      $c_{\text{out}} = \frac{\sigma \cdot 0.5}{\sqrt{0.5^2 + \sigma^2}}$
      $c_{\text{in}} = \frac{1}{\sqrt{0.5^2 + \sigma^2}}$
      $c_{\text{noise}} = 0.25 \cdot \log \sigma$
      Compute network prediction: $D_\theta(\boldsymbol{a}_t, \sigma, g_\phi(\boldsymbol{o}))$
      Compute masked loss: $\mathcal{L} = ((1 - \boldsymbol{b}) \cdot (D_\theta(\boldsymbol{a}_t, \sigma, g_\phi(\boldsymbol{o})) - \boldsymbol{a}_0))^2$
      Apply EDM weighting: $\mathcal{L}_{\text{final}} = \left(\mathcal{L} \cdot \frac{0.5^2 + \sigma^2}{(\sigma \cdot 0.5)^2} \right).mean()$
      Update parameters using Adam ($\eta = 3 \times 10^{-4}$)
      Update EMA model parameters (rate $= 0.999$)
    end for
  end for
end procedure
procedure INFERENCE(Observation $\boldsymbol{o}$, Action mask $\boldsymbol{b}$, Sampling steps $S = 5$)
  Normalize observations using stored statistics
  Initialize $\boldsymbol{a}_S \sim \mathcal{N}(0, 80.0^2 \cdot I)$
  Compute noise schedule: $\sigma_i = \left(0.002^{1/7} + \frac{i}{S}(80.0^{1/7} - 0.002^{1/7})\right)^7$ for $i \in \{0, 1, ..., S\}$
  for $i = S$ down to 1 **do**
    Apply action mask: $\boldsymbol{a}'_i = (1 - \boldsymbol{b}) \cdot \boldsymbol{a}_i$
    Compute network prediction with preconditioning
    ODE update: $\dot{\boldsymbol{a}} = \frac{\boldsymbol{a}_i - D_\theta(\boldsymbol{a}'_i, \sigma_i, g_\phi(\boldsymbol{o}))}{\sigma_i}$
    Euler step: $\boldsymbol{a}_{i-1} = \boldsymbol{a}_i - \dot{\boldsymbol{a}} \cdot (\sigma_i - \sigma_{i-1})$
  end for
  Denormalize using stored statistics
  return $\boldsymbol{a}_{\text{final}}$
end procedure

A.1 Hyperparameters and Training Hardware

The DiT model configuration and training parameters are listed in Table. 3. Our implementation builds on the open-source *CleanDiffuser* library [34], which provides a modular and extensible interface for diffusion models in decision-making tasks. We adopt its implementation of EDM sampling for both training and inference. If the model is trained on a single NVIDIA 3090 GPU, it costs total training time ranging from 3 to 12 hours depending on robot embodiment and dataset size.

A.2 Details of Training and Inference

Our diffusion-based policy handles diverse embodiments through a unified framework that accommodates variable-dimensional observation and action spaces. During training, we normalize all inputs using quantile-based MinMax normalization with 5th and 95th percentiles to mitigate the effects of outliers. This maps features to a common [-1, 1] range while preserving the relative scaling of the majority of data points.

To support cross-embodiment generalization, we zero-pad observation and action vectors to match the largest dimension across all robot morphologies (observations: 68D, actions: 20D) and the dimension configurations of . During both training and inference, we apply morphology-specific binary masks $\boldsymbol{b}$ to ensure only valid dimensions contribute to the loss computation and prediction. Specifically, during training, the loss is computed as: $\mathcal{L} = ((1 - \boldsymbol{b}) \cdot (D_\theta(\boldsymbol{a}_t, \sigma, g_\phi(\boldsymbol{o})) - \boldsymbol{a}_0))^2$

During inference, we employ the Euler method with 5 sampling steps to solve the ODE and generate actions. At each denoising step, we apply the action mask to ensure prediction consistency: $\boldsymbol{a}_i' = (1 - \boldsymbol{b}) \cdot \boldsymbol{a}_i$.

Robot	Observation	Action	Command	Components
Point-Foot Biped	$\mathcal{O}_1 \in \mathbb{R}^{26}$	$\mathcal{A}_1 \in \mathbb{R}^6$	$\mathcal{C}_1 \in \mathbb{R}^3$	Base orientation (3D), angular velocity (3D), joint states (12D), gait phase (2D)
Wheeled Biped	$\mathcal{O}_2 \in \mathbb{R}^{28}$	$\mathcal{A}_2 \in \mathbb{R}^8$	$\mathcal{C}_2 \in \mathbb{R}^3$	Base orientation (3D), angular velocity (3D), joint states (12D), wheel velocities (2D)
Humanoid	$\mathcal{O}_3 \in \mathbb{R}^{68}$	$\mathcal{A}_3 \in \mathbb{R}^{20}$	$\mathcal{C}_3 \in \mathbb{R}^3$	Base orientation (3D), angular velocity (3D), joint states (40D), gait phase (2D)
Quadruped	$\mathcal{O}_4 \in \mathbb{R}^{44}$	$\mathcal{A}_4 \in \mathbb{R}^{12}$	$\mathcal{C}_4 \in \mathbb{R}^3$	Base orientation (3D), angular velocity (3D), joint states (24D), gait phase (2D)

Table 4: Configuration-Specific Observation and Action Spaces

A.3 Ablation Study: Diffusion Sampling Steps

We evaluate how the number of diffusion sampling steps impacts control performance. Fewer steps reduce computation time but may degrade trajectory quality. Results (Fig 6) show that using [3, 5,10,15] denoising steps balances control accuracy and sampling efficiency, with diminishing returns beyond 20 steps. According to this results, we choose 5 denoising steps in the deployment experiments.

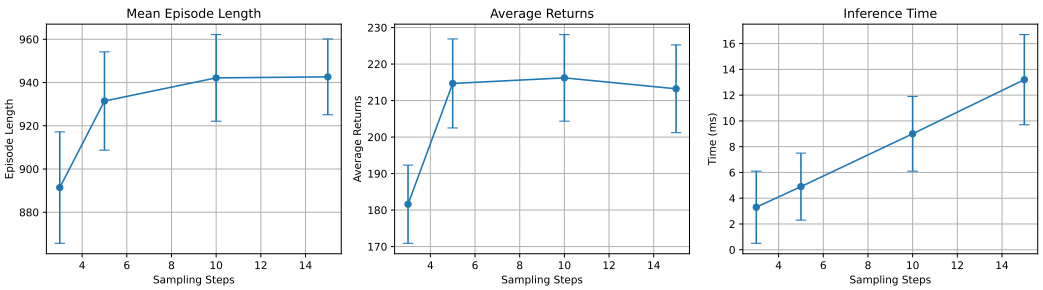


Figure 6: Comparison with Different Sampling Steps in EDM

A.4 Ablation Study: Dataset Size for Diffusion Training

To assess data efficiency, we train diffusion models on varying dataset sizes (1%, 10%, 25%, 50%, 100%) of biped robot while keeping others unchanged. The performance scales sublinearly with data, suggesting strong generalization even with partial data. However, extremely small datasets ($< 25\%$) lead to unstable behaviors and poor terrain negotiation.

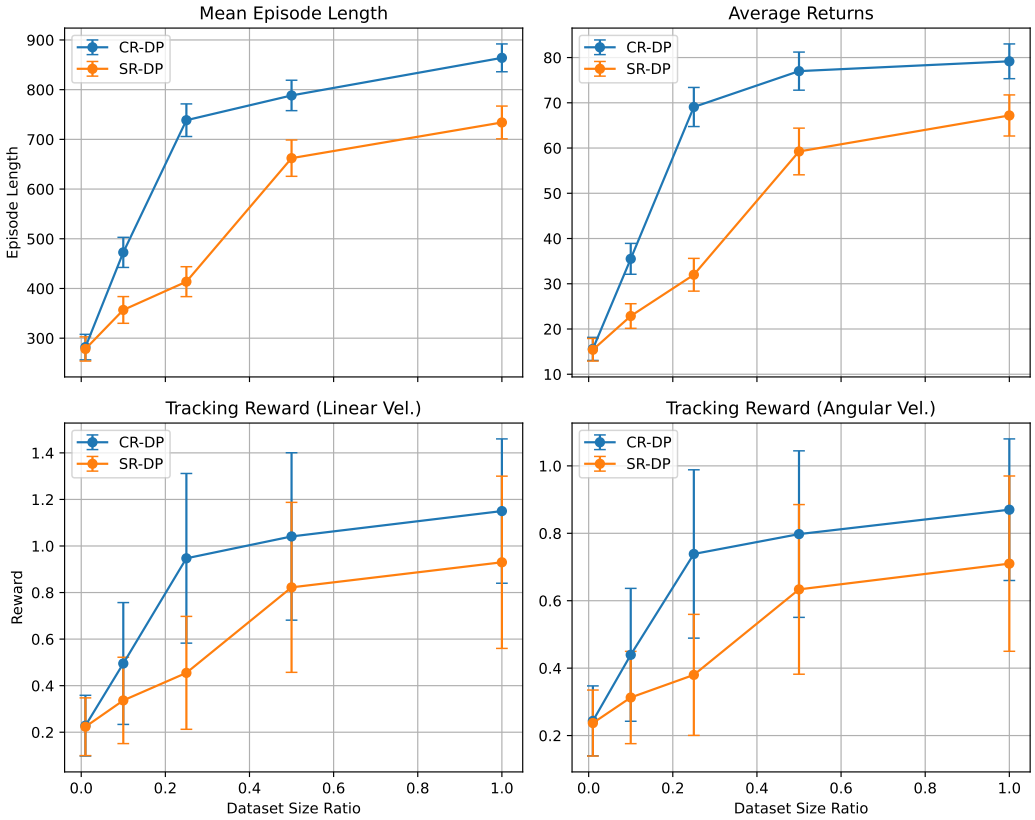


Figure 7: Biped Statistics: Comparison with EDMs Trained with Different Datasets Size. (The maximum number of samples of bipedal robots in the dataset is 2048000)

A.5 Ablation Study: Dataset Composition Analysis for Diffusion Training

We conducted a systematic investigation into how varying data distribution ratios across robot morphologies impact policy performance while keeping the total dataset size constant. Using a baseline configuration with equal 25%-25%-25%-25% allocations for four robot types (point-foot biped, wheeled biped, humanoid, and quadruped), we created four experimental conditions by sequentially reducing one category to 10% while proportionally increasing others to 30%. The resulting Average Return (AR) and Mean Episode Length (MEL) metrics were subsequently analyzed in Fig 8.

Reducing data allocation for pointed-foot bipeds and quadrupeds showed negligible performance impacts, suggesting their relatively simple locomotion tasks require minimal training data. Conversely, decreasing wheeled biped data caused significant performance degradation in its corresponding policy, likely due to the inherent complexity of wheeled locomotion combined with limited cross-morphology knowledge transfer from more dissimilar robot types. Most notably, humanoid data reduction demonstrated dual impacts - not only expected self-performance deterioration from reduced training on its high-dimensional observation space, but also an unexpected AR decrease in wheeled biped performance. This cross-domain dependency confirms the hypothesis presented in our main text regarding humanoid data's complementary role in enhancing wheeled locomotion capabilities, suggesting previously unrecognized synergies between morphologically distinct systems.

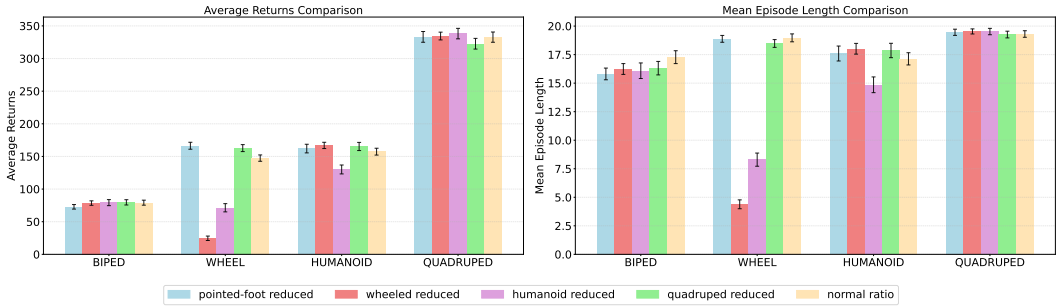


Figure 8: Impact of dataset composition ratios on diffusion policy training. The baseline normal ratio configuration (25%-25%-25%-25%) equally distributes data across four morphologies: pointed-foot biped, wheeled biped, humanoid, and quadruped. Four experimental variations were created by reducing one morphology's share to 10% while proportionally increasing others to 30% each: Pointed-foot reduced (10%-30%-30%-30%), Wheeled reduced (30%-10%-30%-30%), Humanoid reduced (30%-30%-10%-30%), and Quadruped reduced (30%-30%-30%-10%).

A.6 Zero-Shot Transfer to an Unseen Platform: Unitree Go2

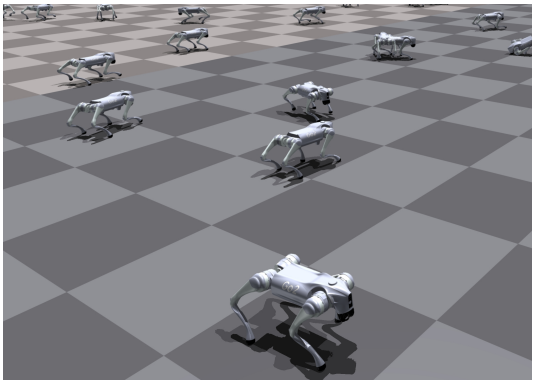


Figure 9: Zero-Shot Transfer to Unitree Go2

To evaluate generalization beyond training morphologies, we directly deploy the unified policy on Unitree Go2 without any finetuning. Despite being unseen during training, the policy achieves stable forward locomotion, demonstrating the model's morphology-agnostic generalization. Detailed results and rollout videos are available in the supplementary video.

For locomotion task evaluation, the Mean Episode Length (MEL), Locomotion Velocity Tracking (LVT), and Actuator Variation Threshold (AVT) metrics quantify task completion quality. As Table 5 indicates, the statistically insignificant differences of CR-DP+RA in MEL ($0.62 \sim 3.2\%$), LVT ($0.22 \sim 4.1\%$), and AVT ($0.23 \sim 5.3\%$) between configurations a1 and go2 confirm successful zero-shot policy transfer to the go2 morphology. Also, the simulation outcomes illustrated in Figure 9 confirm successful zero-shot transfer to the go2 configuration, evidencing strong generalization capability of our policy framework. While Table 5 reveals a substantial disparity in Average Return (AR) between CR-DP+RA implementations for a1 (387.57) and go2 (295.96), our analysis identifies this discrepancy as primarily attributable to differing desired base height parameters in the base height tracking reward module. Due to morphological variations between configurations, the optimal base height settings should be configuration-specific. After normalizing for this parameter by subtracting the base height reward component, the adjusted AR values for CR-DP+RA become 306.69 (a1) and 294.28 (go2), indicating comparable performance when accounting for morphological differences. This adjustment validates that the observed AR gap stems from parameter specification rather than fundamental policy limitations.

Group	Method	Metrics			
		AR↑	**MEL↑**	**LVT↑**	**AVT↑**
a1	CR-DP+RA	387.57±6.15	19.60±0.19	5.42±0.41	4.32±0.41
	CR-DP	325.24±8.63	19.14±0.36	5.22±0.72	4.14±0.58
go2	CR-DP+RA	295.96±5.81	18.98±0.33	5.20±0.74	4.09±0.60
	CR-DP	256.05±5.72	18.40±0.37	5.02±0.91	3.93±0.74

Table 5: Performance Comparison of quadruped a1 and go2

B RL residual adaptation Detail

While diffusion models provide strong priors for action generation, they may lack task-awareness or fail to account for nuanced terrain interactions. To complement the generative prior, we introduce a residual policy trained via RL, detailed below.

Our implementation leverages NVIDIA IsaacGym's GPU-accelerated parallel simulation environment specifically designed for robotic learning. The system architecture integrates with the established from `humanoid_gym` [1] and `legged_gym` [31] open-source frameworks, implementing a scalable reinforcement learning pipeline based on Proximal Policy Optimization (PPO).

Baseline policies were trained from scratch using 10k PPO iterations (50k epochs), which we found sufficient for convergence in most cases. However, we agree that our unified policy benefits from more total samples (due to the pretraining stage), and this may introduce a little comparison bias.

B.1 Environment Setup

The policy observations $\boldsymbol{o}$ for the robot include velocity control commands and proprioceptive sensory data with gait cycle parameters (excluding wheeled locomotion) in the past 10 steps. Specifically:

- Proprioceptive Data:
  - Robot pose (orientation) and angular velocity measured by the IMU.
  - Joint angles and angular velocities of all robotic limbs.
- Velocity Control Commands:
  - Linear velocity commands in the XY-plane (Cartesian coordinates).
  - Angular velocity command around the Z-axis (yaw direction).
- Gait Cycle:
  A time-dependent periodic signal defined by parametric sinusoidal curves:
  $$\text{Gait}(t) = \begin{cases} \sin\left(2\pi t/T\right) \\ \cos\left(2\pi t/T\right) \end{cases}$$
  where T is the gait period (time to complete one cycle),t represents the current time step.

The PPO hyperparameters are detailed in Table 6, with a notable configuration choice of disabling value prediction clipping. This design decision stems from our hybrid architecture that integrates a pretrained diffusion model as prior guidance. The pre-trained dynamics awareness enables more stable value function initialization compared with conventional scratch training paradigms.

PPO Parameter	**Value**
Desired KL	0.01
Learning Rate	4e-4
Discount Factor	0.99
Lambda(GAE)	0.95
Mini Batches	4
Learning Epochs	5
Entropy Loss Scale	0.001
PPO Clip Range	0.2
Values Predicted Clip	False
Residual Coeff	0.2
Max Iterations	10001
Rollouts	24

Table 6: Training Parameters for PPO Unified EDM

Table 7 presents our heterogeneous actor-critic architecture featuring a unified actor network parameters and four specialized critic network parameters.

Model - Critic Biped	
Activations	`["elu", "elu", "elu", "linear"]`
Hidden Dims	`[512, 256, 128]`
Model - Critic Biped Wheel	
Activations	`["elu", "elu", "elu", "linear"]`
Hidden Dims	`[512, 256, 128]`
Model - Critic Humanoid	
Activations	`["elu", "elu", "elu", "linear"]`
Hidden Dims	`[512, 256, 128]`
Model - Critic Quadruped	
Activations	`["elu", "elu", "elu", "linear"]`
Hidden Dims	`[512, 256, 128]`
Model - Actor	
Log Std Max	4.0
Log Std Min	-20.0
Std Init	1.0
Activations	`["elu", "elu", "elu", "linear"]`
Hidden Dims	`[512, 256, 128]`

Table 7: Actor-Critic Model Parameters

B.2 Reward Design

Our reward function comprises three coordinated components systematically designed for robust policy learning:

- Task-specific objectives governing locomotion performance, shown in Table 8
- Morphology-aware regularization terms addressing physical constraints , shown in Table 9
- A diffusion-guided residual penalty enforcing dynamic feasibility through pretrained motion priors:

$$r_d(\Delta \boldsymbol{a}_t) = \alpha \|\Delta \boldsymbol{a}_t\|_1$$

where α is the reward coefficient of residual penalty, $\Delta \boldsymbol{a}_t$ is the residual action.

Reward	Expression	Biped-Wheel	Pointed-Foot-Biped	Humanoid	Quadruped
Tracking Linear Velocity	$\exp(-\|v_b^{des} - v_b\| \times \sigma)$	4.0	2.0	2.0	6.0
Tracking Angular Velocity	$\exp(-\|\Omega_b^{des} - \Omega\| \times \sigma)$	2.0	1.5	1.5	5.0
Base Height	$\exp(-\|h_b^{des} - h_b\| \times 100)$	-	2.0	4.0	6.0
Orientation	$\exp(-\|\theta_b^{des} - \theta_b\| \times 10)$	5.0	5.0	-10.0	4.0

Table 8: Locomotion Task Reward

Reward	Expression	Biped-Wheel	Pointed-Foot-Biped	Humanoid	Quadruped
Joint Torque	$\|\boldsymbol{\tau}\|^2$	-1.6e-4	-8e-5	-8e-5	-2e-4
Power	$\|\tau \cdot \dot{\boldsymbol{q}}\|$	-2e-5	-2e-5	-	-5e-4
Joint Vel	$\|\dot{\boldsymbol{q}}\|^2$	-5e-4	-	-	-
Joint Acc	$\|\ddot{\boldsymbol{q}}\|^2$	-1.5e-7	-2.5e-7	-2.5e-7	-2.5e-7
Linear Velocity Z	$\|v_b^z\|^2$	-0.3	-0.5	-2.0	-2.0
Angular Velocity XY	$\|\Omega_{xy}\|^2$	-0.3	-0.05	-0.05	-0.1
Action Smoothness	$\|a_{k+1} + a_{k-1} - 2a_k\|^2$	-0.03	-0.01	-0.01	-0.02
Action Rate	$\|a_{k+1} - a_k\|^2$	-0.03	-0.01	-0.01	-0.02
Collision	$\|\boldsymbol{F}_c\| > 0.$	-0.1	-0.02	-1.0	-10.0
Contact Force	$\text{clip}\{\|\boldsymbol{F}_l\|_2 + \|\boldsymbol{F}_r\|_2 - F_{max}\}_0^{400}$	-0.1	-0.1	-	
Default Joint Position	$\|\boldsymbol{q} - \boldsymbol{q}_0\|$	-0.05	-	3.0	2.0
Foot Distance	$\text{clip}(\boldsymbol{d}_{foot}^{min} - \boldsymbol{d}_{foot})$	-100	-100	-100	-
Nominal Foot Height	$\exp(-\|h_{foot}^{des} - h_{foot}\|^2 \times 200)$	4.0	-	-	3.0

Table 9: Regularization Reward