Towards Embodiment Scaling Laws in Robot Locomotion

Bo Ai[1*] Liu Dai[1*] Nico Bohlinger[4*] Dichen Li[1*] Tongzhou Mu[1]
Zhanxin Wu[3] K. Fay[1] Henrik I. Christensen[1] Jan Peters[4,5] Hao Su[1,2]
*Equal contribution
[1]University of California San Diego, USA [2]Hillbot Inc., USA
[3]Cornell University, USA [4]Technical University of Darmstadt, Germany
[5]German Research Center for AI (DFKI); Robotics Institute Germany; hessian.AI, Germany

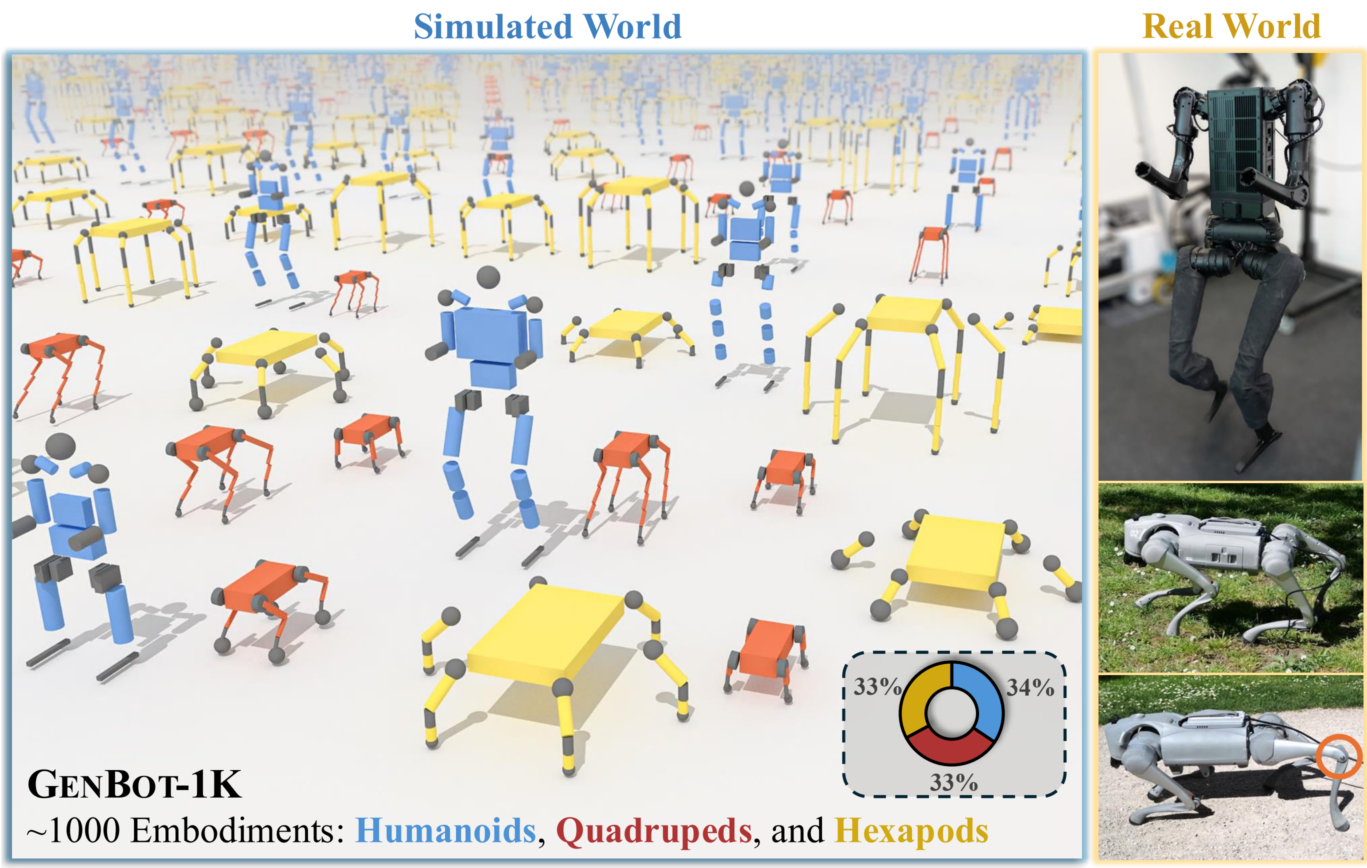


Fig. 1: **One Policy, Two Worlds, Many Robots.** We study embodiment scaling laws by training a single policy on ∼1,000 procedurally generated "blueprint" embodiments in simulation. Our policy zero-shot transfers to real-world embodiments, including modified joint constraints (circled in red).

Abstract—**Developing generalist agents that can operate across diverse tasks, environments, and robot embodiments is a grand challenge in robotics and artificial intelligence. In this work, we focus on the axis of embodiment and investigate *embodiment scaling laws*—the hypothesis that increasing the number of training embodiments improves generalization to unseen ones. Using robot locomotion as a test bed, we procedurally generate a dataset of ∼1,000 varied embodiments, spanning humanoids, quadrupeds, and hexapods, and train generalist policies capable of handling diverse observation and action spaces on random subsets. We find that increasing the number of training embodiments improves generalization to unseen ones, and scaling embodiments is more effective in enabling embodiment-level generalization than scaling data on small, fixed sets of embodiments. Notably, our best policy, trained on the full dataset, zero-shot transfers to novel embodiments in the real world, such as Unitree Go2 and H1. These results represent a step toward general embodied intelligence, with potential relevance to adaptive control for configurable robots, co-design of morphology and control, and beyond.**

I. Introduction

Over two millennia ago, Heraclitus remarked that no man ever steps in the same river twice. Today, one might say that no embodied agent acts in the same body twice. In a broad sense, the human embodiment evolves in subtle ways, e.g., through injury, aging, or tool use [62, 69, 100], which alter our sensorimotor coordination. In robotic systems, similar variability in embodiment can arise due to manufacturing differences, hardware upgrades, or mass deployment of heterogeneous robots. How can we learn policies that can zero-shot transfer across a large number of distinct embodiments?

Scaling has been a key driver of progress in deep learning, which can occur along multiple **dimensions**. Scaling **dataset size** and **model size** has improved generalization in vision [13, 50, 61, 71, 91, 93, 99, 104] and language [1, 3, 20, 21, 30, 31, 38, 46, 72, 101]. In robotics, scaling the number of **tasks** [10, 26, 27, 33, 41, 53, 70, 110] and **environments** [2, 4, 23, 25, 32, 33, 41, 56, 96, 110] enable cross-task and cross-environment generalization. In this work, we explore a distinct and underexplored dimension of scaling: **robot embodiment**, the physical structure of robots. We hypothesize that scaling the number of training embodiments leads to better generalization to unseen embodiments, as the policies learn to capture shared control strategies across different physical structures. We refer to this hypothesized relationship as ***embodiment scaling laws***.

Studying this hypothesis requires addressing several open challenges. First, we need policy architectures that (i) can be conditioned on embodiment structures, (ii) handle varied observation and action spaces, (iii) scale to a large number of distinct embodiments, and (iv) have the right inductive biases to discover generalizable motion patterns for zero-shot transfer to novel embodiments. Second, we need a large dataset comprising diverse robot morphologies. We postulate that an order of magnitude of 10^3 is a reasonable starting point to study the relationship between the number of training embodiments and generalization performance on novel ones. However, most prior work on multi-embodiment policy training has not systematically investigated the relationship between generalization performance and the number of training embodiments at scale, likely due to the limited size of available embodiment collections, often restricted to $\sim 10^2$ in simulation [74] or $\sim 10^1$ in the real world [23, 49, 70, 96]. Consequently, the scale, degree of generalization, and specific scaling analysis presented in this work remain largely unexplored.

To this end, we develop a framework for studying embodiment scaling laws using robot locomotion as a testbed. We first use a procedural generation algorithm to create GenBot-1K, a large-scale dataset of ∼1,000 blueprint robot descriptions in the URDF format, including humanoids, quadrupeds, and hexapods. To handle varied state and action spaces, we extend Unified Robot Morphology Architecture (URMA) [8] into a wider multi-head attention architecture. We adopt a two-stage policy learning framework [44, 97]: (i) training single-embodiment expert policies using Reinforcement Learning (RL), and (ii) distilling these experts into a single embodiment-aware URMA policy via behavior cloning. We vary the number of embodiments used in distillation to study the effect of embodiment scaling on embodiment-level generalization.

Overall, we present a large-scale empirical study of embodiment scaling laws across ∼1,000 robot embodiments. We design a general reward formulation, training curriculum, and domain randomization that enable scalable training of embodiment-specific RL experts without embodiment-specific tuning, accumulating a total of 2 trillion simulation steps. We observe a positive correlation between the number of training embodiments and generalization performance on held-out test embodiments. The best policy, trained on 2 billion expert demonstration steps across the full set of training embodiments, achieves zero-shot transfer to real-world robots, including the Unitree Go2 with varied kinematic constraints and the H1 humanoid. These findings provide preliminary empirical evidence for embodiment scaling laws and highlight their potential for enabling generalist robot agents.

II. Related Work

Cross-Embodiment Generalization. One goal of cross-embodiment learning is to enable control policies to generalize across robot embodiments without retraining. Prior efforts often focus on transferring policies between a small number of robots by aligning dynamics, learning shared embeddings [15, 107], or extracting transferable skills [39, 58]. However, these methods are only able to transfer to a single or a few target embodiments. Related work about scalable network architectures, such as graph neural networks [40, 98] or Transformers [29, 94], scale to more complex embodiments by conditioning on embodiment-specific information, but these works mostly use unrealistic and simplified robots that are not suitable for real-world transfer. More recent approaches can be trained on a larger number of realistic robot embodiments, but they often rely on existing low-level controllers [84, 89], embodiment-specific decoders [22], other action abstractions [24, 83], or assume a fixed observation and action space [28], limiting their generalization capabilities to pre-defined morphological structures. URMA [8] solves this issue by introducing a unified joint-level control architecture for arbitrary robot morphologies, but is validated only on 16 robots without studying scaling effects. Our work demonstrates broader cross-embodiment generalization than prior works by training a single policy on ∼1,000 embodiments, achieving zero-shot transfer to unseen embodiments in both simulation and the real world.

Robot Locomotion. In recent years, Deep Reinforcement Learning (DRL) has been applied to single embodiment robot locomotion to great success. The combination of scalable on-policy RL algorithms, such as Proximal Policy Optimization (PPO) [81], with fast and highly parallelizable simulators has enabled the training of powerful locomotion policies for quadruped [11, 16, 19, 64, 67, 90, 108] and humanoid robots [52, 55, 78, 86, 109]. Techniques such as student-teacher learning [14, 48], curriculum learning [51, 63, 80], and domain randomization [12, 51, 76] have enabled zero-shot sim-to-real transfer of these policies. Less data-hungry methods for learning directly on real robots, utilizing model-based or off-policy RL algorithms [9, 54, 87, 88], and non-learning methods, such as Model Predictive Control (MPC) [42, 47], have also been proposed for legged locomotion, but generally

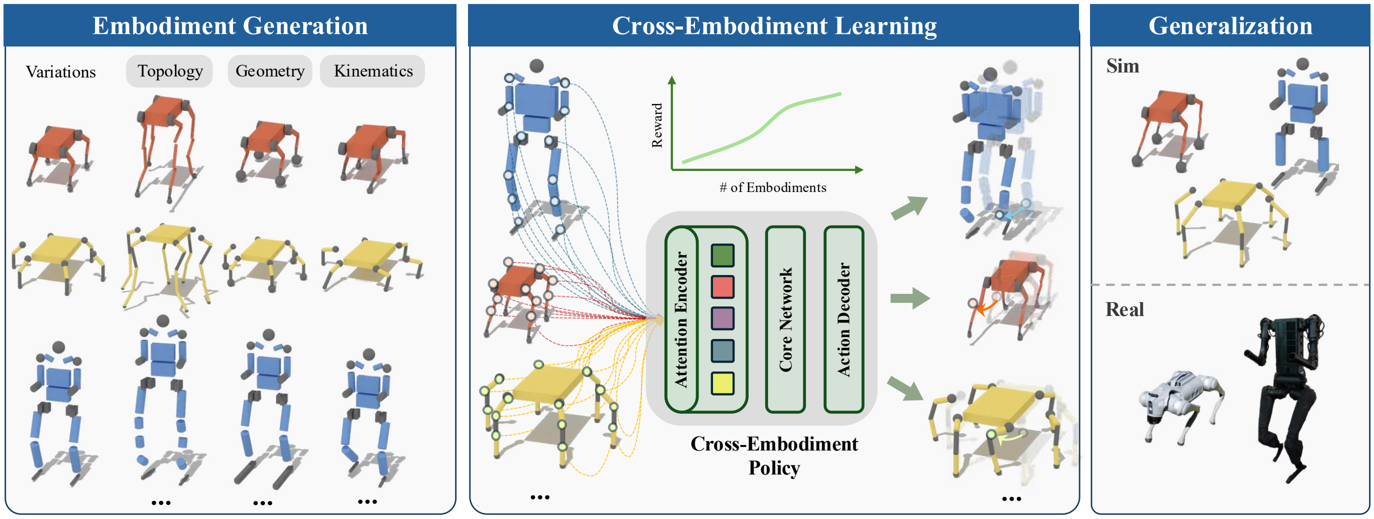


Fig. 2: **Overview of our approach for studying embodiment scaling laws.** We procedurally generate GenBot-1K, a dataset of ∼1000 diverse robot embodiments with structured variations in topology, geometry, and kinematics. We train a single cross-embodiment policy using the URMA architecture, which handles varying observation and action spaces via attention-based joint encoding. We systematically vary the number of training embodiments to study how generalization scales with embodiment quantity. The policy trained on the full training dataset transfers zero-shot to novel simulated robots and real-world hardware with different morphologies.

trade their efficiency for worse performance with less robust gaits on challenging terrain or under strong perturbations.

Robot Embodiment Generation. Prior research in robot embodiment generation has pursued several directions. One prominent direction focuses on optimizing robot designs for specific tasks, where procedural and learning-based techniques generate embodiments tailored for enhanced performance in tasks such as locomotion [5, 79, 106] or manipulation [36]. Closer to our objectives is the use of embodiment generation to develop generalizable robot policies. Existing works have explored methods based on simplified kinematic trees [29, 34], randomization within a fixed morphology [28], diverse sensor configurations [24], or varied hand structures [75]. However, these approaches are generally limited to a single robot class or topological template. In contrast, we introduce a comprehensive procedural generation framework that spans multiple morphological classes, including quadrupeds, hexapods, and humanoids, while varying topology, geometry, and kinematics for each of them. This enables a large-scale systematic study of embodiment scaling in locomotion.

III. Methodology

Generalizable cross-embodiment robot learning aims to train a control policy that can control diverse *seen* and *unseen* robot embodiments to solve a common task. Formally, let $\mathcal{E}$ denote a set of embodiments sampled from $\mathcal{P}_{\mathcal{E}}$, where each embodiment $e \in \mathcal{E}$ is defined as a triplet $e = \langle \mathcal{G}, \mathcal{T}, \mathcal{K} \rangle$, where $\mathcal{T}$ specifies the joint topology (i.e., number and connectivity), $\mathcal{G}$ denotes link geometry (e.g., shape and size), and $\mathcal{K}$ describes additional kinematic properties (e.g., joint types and range of motion). The control problem of each embodiment e is defined by a Markov Decision Process (MDP) $\mathcal{M}_e = \langle \mathcal{S}_e, \mathcal{A}_e, P_e, R_e, H \rangle$, where $\mathcal{S}_e$, $\mathcal{A}_e$, and P_e denote the state space, action space, and transition dynamics; R_e is the reward function; and H is the episode horizon. At any particular time step t, a policy predicts an action $a_t \in \mathcal{A}_e$, conditioned on the robot state $s_t \in \mathcal{S}_e$ and the embodiment descriptor $\phi(e)$. In the specific case of robot locomotion, the policy is additionally conditioned on a x-y-yaw velocity command $v_t \in \mathbb{R}^3$ with respect to the trunk frame, i.e., $a_t \sim \pi(s_t, \phi(e), v_t)$.

During training, we optimize the policy to maximize the expected cumulative reward across training embodiments $\mathcal{E}_{\text{train}} \subset \mathcal{E}$ with trajectories $\tau = \{(s_0, a_0), \ldots, (s_H, a_H)\}$ sampled from $\mathcal{M}_e$:

$$\pi^*_{\text{train}} = \arg\max_{\pi} \mathbb{E}_{e \in \mathcal{E}_{\text{train}}} \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{H} R_e(s_t, v_t, a_t) \right]. \tag{1}$$

The generalization performance is evaluated on a held-out set of embodiments $\mathcal{E}_{\text{test}} = \mathcal{E} \setminus \mathcal{E}_{\text{train}}$:

$$J_{\text{test}}(\pi^*_{\text{train}}) = \mathbb{E}_{e \in \mathcal{E}_{\text{test}}} \mathbb{E}_{\tau \sim \pi^*_{\text{train}}} \left[\sum_{t=0}^{H} R_e(s_t, v_t, a_t) \right]. \tag{2}$$

We note that both *learning* and *generalizing* across embodiments present significant challenges. Differing observation and action spaces require policies to handle variable-sized and potentially inconsistent inputs and outputs. Variations in kinematic constraints, self-collision profiles, and contact dynamics introduce embodiment-specific behaviors that complicate the optimization landscape of policy learning. Even further, generalizing to unseen embodiments demands that the policy captures meaningful shared control features that can be applied to novel physical embodiments.

Scaling hypothesis. We hypothesize that generalization improves with the number of training embodiments, i.e., larger $|\mathcal{E}_{\text{train}}|$ leads to higher J_{test}. Intuitively, training on

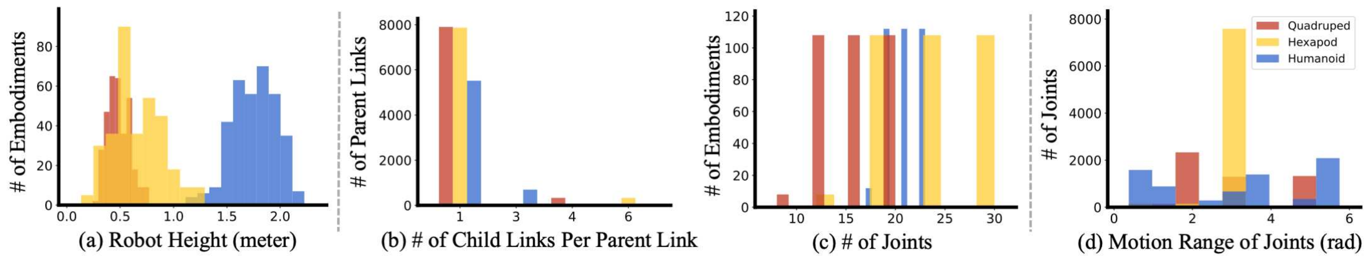


Fig. 3: **Empirical distributions of embodiment variations in GENBOT-1K.** The statistics reflect geometric (a), topological (b,c), and kinematic (d) variability of embodiments in our dataset.

more diverse embodiments encourages the policy to extract structural features that transfer to novel robots. For instance, despite differences in leg length or joint placement, many embodiments share similar locomotion dynamics and constraints. Discovering a scaling trend would provide empirical support for an embodiment scaling law and offer actionable insights for building general-purpose control policies.

Empirical setup. To study the hypothesis, we fix a constant test set by randomly holding out 20% of the generated embodiments. The remaining 80% serve as the pool for constructing training subsets $\mathcal{E}_{\text{train}}^{(i)} \subset \mathcal{E}_{\text{train}}$ at varying proportions $i \in (0, 1]$. For each subset, we train a separate policy $\pi_{\text{train}}^{(i)*}$ and evaluate it on the fixed $\mathcal{E}_{\text{test}}$. This setup enables a systematic analysis of generalization performance $J_{\text{test}}(\pi_{\text{train}}^{(i)*})$ as a function of training set size, probing for evidence of an embodiment scaling law.

Next, we describe how we generate diverse embodiments (Section III-A), construct a policy to handle varying observation and action spaces (Section III-B), and train it on many embodiments (Section III-C).

A. Embodiment Generation

We adopt a procedural generation pipeline to produce diverse robot embodiments spanning three commonly used morphology classes: humanoid [7, 8, 17, 43, 52, 82, 85], quadruped [8, 16, 37, 51, 57, 63, 65, 83, 92, 102], and hexapod [6, 8, 18, 35, 73, 77, 103, 105]. Our generated robots follow common design patterns using realistic base components, such as link shapes, dimensions, and motor properties, but are procedurally composed into novel embodiments by varying their parameters. Geometric variation is introduced by scaling individual links and overall body size. Topological variation is achieved by changing the number of knee joints per leg within each morphology class. We also vary joint limits to implement kinematic variations. In total, we generate 1,012 distinct robots, including 348 humanoids, 332 quadrupeds, and 332 hexapods, to form the GENBOT-1K dataset (Figure 1). Our resulting dataset is diverse in various aspects, as reflected in post-generation statistics (Figure 3). More details about the generation process are provided in Appendix VII.

B. Cross-Embodiment Policy Architecture

To train a policy that can control $\sim$1000 different embodiments with different state and action spaces, we use URMA, an embodiment-aware architecture for robots with arbitrary numbers of joints [8]. URMA handles the differently sized partially observable states (observations) o of different embodiments by splitting them into fixed-length general observations o_g and varying-length joint-specific observations o_j, depending on the number of joints $j(e)$. The embodiment descriptors $\phi(e)$ are used to generate joint description vectors d_j, which can uniquely describe every joint of the embodiment and are made up of the fixed dynamics and kinematics properties of the joint and its underlying motor. The joint-specific observations are processed by an attention encoder and are summed up into the joint latent vector

$$\bar{z}_{\text{joints}} = \sum_{j \in J} z_j, \qquad z_j = \frac{\exp\left(f_\phi(d_j)/\tau\right)}{\sum_{L_d} \exp\left(f_\phi(d_j)/\tau\right)} f_\psi(o_j), \quad (3)$$

where f_ϕ (with latent dimension L_d) and f_ψ are the encoders for the joint descriptions and joint observations, respectively, and τ is the learnable temperature parameter of the softmax. Intuitively, the attention mechanism fuses joint observations based on their descriptions so that $\bar{z}_{\text{joints}}$ has global information about the embodiment. The encoded joint latent vector is then concatenated with the general observations and processed by a core network to generate an action latent vector $\bar{z}_{\text{action}} = h_\theta(o_g, \bar{z}_{\text{joints}})$. To handle the differently sized action spaces, URMA concatenates the action latent vector with each encoded joint description vector in batch to decode a single action for each joint:

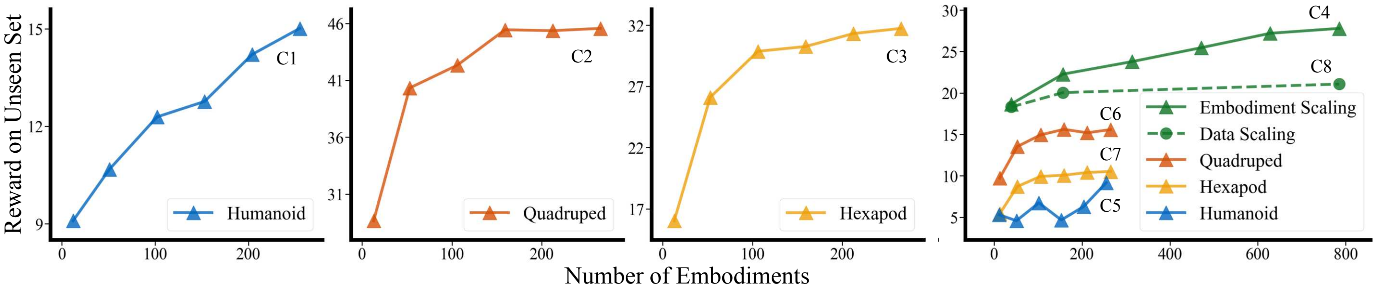
$$a_j = \mu_\nu(g_\omega(d_j), \bar{z}_{\text{action}}, z_j), \quad (4)$$

where g_ω is the action encoder for the joint descriptions, μ_ν is the final action decoder. In our work, we incorporate multi-head attention into URMA, enabling the policy to attend to different joint-level features in parallel and better capture complex inter-joint dependencies (Appendix VIII-B).

C. Two-Stage Policy Learning

To scale cross-embodiment policy learning to a large number of robots, we adopt a two-stage paradigm. First, we train embodiment-specific expert policies using standard RL. Then, we collect demonstration data from these experts and train a single student policy via imitation learning, conditioned on embodiment descriptors. This approach allows learning across $\sim$1000 robots while maintaining tractable memory usage and stable training dynamics.

Expert Training. We develop a unified RL locomotion training pipeline applicable to all embodiments with minimal



(a) Embodiment Scaling within Morphology Classes (b) Cross-Class Learning

Fig. 4: **Results on embodiment scaling.** We evaluate generalization performance as a function of the number of training embodiments. (a) In-class study: policies are trained and tested within the same morphology class (humanoid, quadruped, or hexapod). (b) Cross-class study: We train policies on the full training set (green) and compare performance against policies trained on only the individual classes, while all policies are evaluated on the test set containing all classes. The proportion of training embodiments ($i \in \{0.05, 0.2, 0.4, 0.6, 0.8, 1.0\}$) is denoted in the x-axis. While the underlying reward function is the same, the reward scales differ across classes due to inherent differences in the embodiments (e.g., humanoids are less stable than quadrupeds) and unnormalized reward formulations (e.g., humanoids experience larger ground contact forces).

tuning. Key components include extensive domain randomization, performance-based curriculum learning, and regularization terms that encourage stable and natural locomotion (e.g., penalizing jittering movements and excessive ground contact). All robots in one morphology class share one set of hyperparameters for scalable training.

Training robust policies for $\sim$1,000 robot embodiments is computationally demanding. We use NVIDIA Isaac Lab [68], a GPU-accelerated simulation framework, to train single-embodiment policies across 4096 parallel environments with PPO [81]. Training all experts takes approximately 5 days on 160 NVIDIA RTX 4090/3090 GPUs, totaling over 2 trillion simulation steps. Full details on the training process are provided in Appendix VI.

Student distillation. Given expert policies $\{\pi_e\}_{e\in\mathcal{E}_{train}}$, we collect a demonstration dataset by rolling out each policy for 600 timesteps in 4,096 parallel environments, totaling 2 billion samples across all embodiments. We then train URMA by minimizing the Mean Squared Error (MSE):

$$\mathcal{L}_{\text{BC}} = \mathbb{E}_{(s_t,e,a_t)\sim\mathcal{D}}\left[\|\pi(s_t,\phi(e)) - a_t\|^2\right], \quad (5)$$

where $\mathcal{D}$ is the expert demonstration dataset. The student policy conditions on the embodiment descriptor $\phi(e)$, enabling it to generalize across the generated embodiments with different geometry, topology, and kinematics. Training the model on the full demonstration dataset takes one week using a NVIDIA H100 GPU. More details about the distillation process can be found in Appendix VIII.

IV. Experiments

In this section, we conduct a large-scale empirical study to investigate the scaling behavior of cross-embodiment learning. Our experiments are designed to answer the following key research questions:

Q1. How does the generalization performance of the cross-embodiment policy scale with the number of training embodiments? (Sec. IV-A)

Q2. Can the learned policy generalize zero-shot to unseen embodiments, including real-world robots, and handle varied kinematic constraints? (Sec. IV-B)

Q3. Does the policy network learn meaningful, structured representations of the space of robot embodiments and morphologies through cross-embodiment training? (Sec. IV-C)

A. Studying Embodiment Scaling Laws

We train and evaluate our policies under multiple setups and show the results in Figure 4. We analyze the generalization patterns through the three aspects below.

Scaling within each embodiment class. We conduct training and evaluation separately for each morphology class (humanoid, quadruped, and hexapod) in GenBot-1K, resulting in curves C1–C3. For each morphology class, we observe a clear scaling trend: increasing the number of training embodiments from 0.05 to 1.0 can double the cumulative reward. The rate of convergence varies by class: for quadrupeds and hexapods, performance saturates around 100 training embodiments, while for humanoids, it continues to improve steadily with more training data, likely due to greater instability and control difficulty. This suggests that more challenging embodiments may benefit more from larger-scale embodiment scaling.

Scaling across embodiment classes. We train on the full combined dataset of all three classes and evaluate on a unified test set (C4). The resulting curve begins at a reward of 18 and rises consistently to nearly 30, demonstrating that scaling across diverse embodiments enables broader generalization. We further evaluate the policies trained on individual morphology classes (corresponding to C1–C3) on the combined test set, obtaining (C5–C7). Since each of these models has only seen a single morphology class during training, their performance on the mixed test set is limited. In contrast, the best point on C4 achieves 2–5$\times$ higher average reward than C5–C7, demonstrating that training across diverse morphology classes enables substantially broader generalization.

Comparison with pure data scaling. To disentangle the effects of embodiment diversity from data quantity, we collect a dataset using only 5% of the training embodiments and

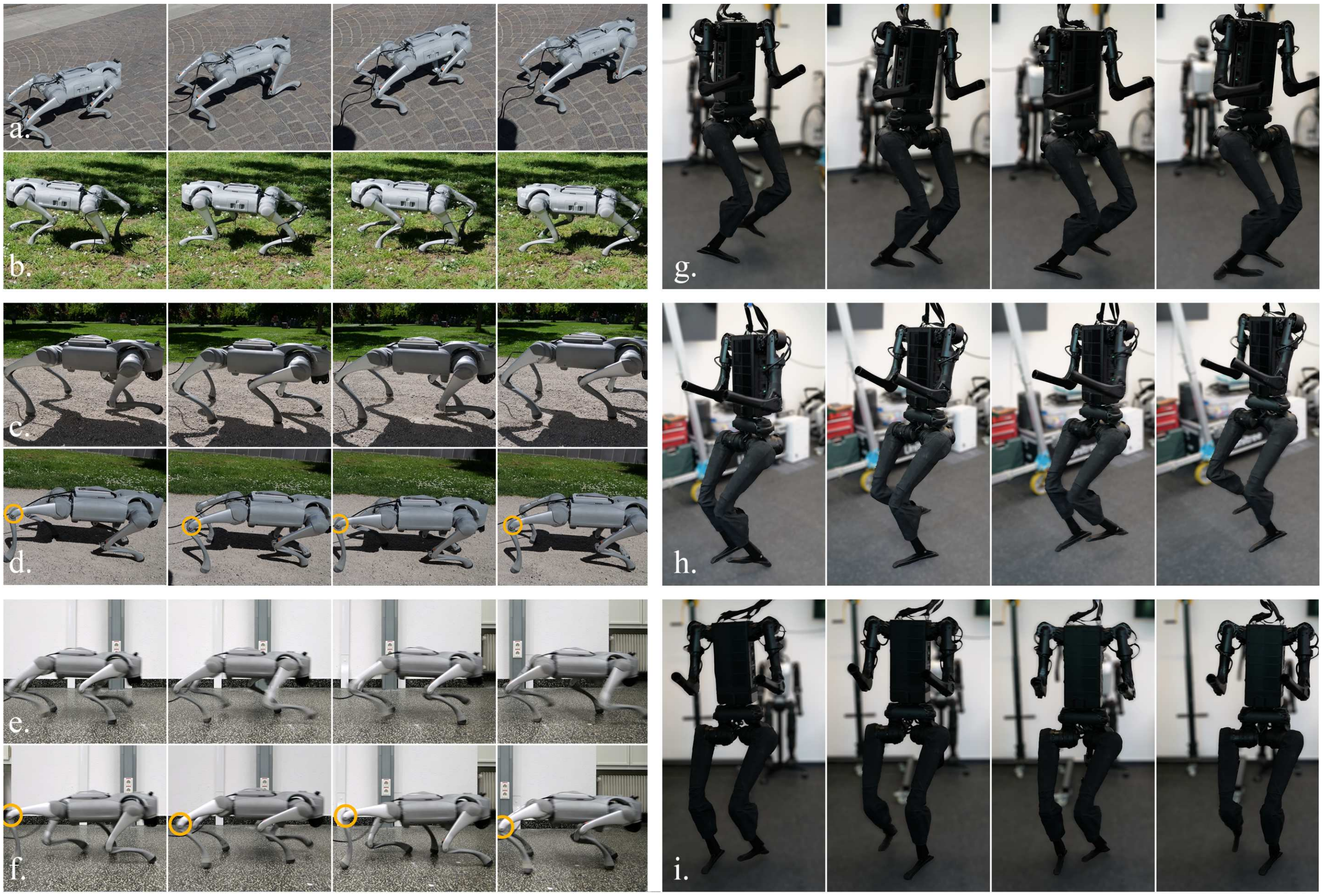


Fig. 5: **Zero-shot generalization to unseen real-world robots.** Our URMA policy, trained on 817 diverse simulated embodiments, successfully transfers zero-shot to control the Unitree Go2 quadruped and Unitree H1 humanoid in the real world. **a, b:** The policy can perform forward and backward locomotion on cobblestone and grass terrain with the Go2. **c, d, e, f:** We test the policy's adaptation to kinematic constraints by artificially restricted the joint limits on the right rear knee of the Go2 by 20%. The policy effectively compensates for the limited range of motion, resulting in a stable limping gait on gravel (d) and indoors (f), compared to the unrestricted gait (c, e). **g, h:** Zero-shot transfer on the H1 works well in a lab environment, showing decent forward and backward locomotion. **i:** Walking side-to-side with H1 is slower as in simulation but stable in the real world.

vary the number of trajectories per embodiment for distillation (C8). We find that performance quickly saturates: the policy nearly reaches its peak at 0.2 data scale (4× data as 0.05), with negligible gains beyond that. This highlights that, if the goal is to achieve broad embodiment-level generalization, it is ineffective to only increase data volume on a small set of embodiments; embodiment scaling is essential.

B. Real-World Generalization Test

To validate real-world transfer capabilities, we conducted zero-shot deployments of our best-performing policy, trained on the full training set of 817 simulated embodiments, on two real robots: the Unitree Go2 quadruped and the Unitree H1 humanoid, neither of which was included in the training set $\mathcal{E}_{\text{train}}$, although robots with similar kinematic structures were present.

Figure 5 shows the policy successfully generalizing to the two real robots without any fine-tuning or modifications, using only the URDF of the respective robot to generate the embodiment descriptor $\phi(e)$. The Go2 demonstrated robust and stable walking gaits across diverse terrains such as grass, cobblestone, and gravel (a-c). Similarly, the H1 was able to maintain stable locomotion, tracking the desired velocity commands while walking on flat ground with rubber mats in a lab environment (g-i). While the transfer worked for both robots, the policy transferred worse to the H1 compared to the Go2, highlighting the need for potentially even more diverse humanoid robots in the training set.

To probe the policy's ability to handle kinematic variations in the real world, we artificially restricted the joint limits of the knee joints of the Go2 to 20% of their nominal limits by restricting the PD controller on the robot and pushing towards the limits with high gains when the joint angles exceed the limits. Figure 5 (d, f) shows that the policy was able to transfer the adaptations it learned in simulation as it keeps the restricted

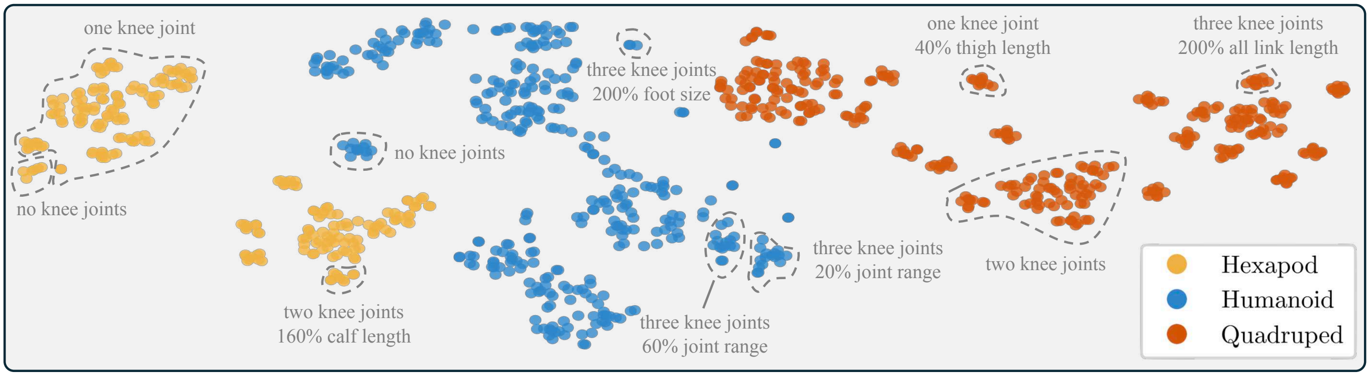


Fig. 6: **Visualization of the embodiment latent space.** t-distributed Stochastic Neighbor Embedding (t-SNE) projection of the action latent vectors on the complete GENBOT-1K dataset from the URMA policy trained on the full training set. Points are colored by morphology class. The clear clustering based on morphology class, and finer sub-structures related to the number of knee joints or kinematic and geometric properties, suggest that the policy learns a meaningful and structured representation of diverse embodiments, capturing functional similarities that help during cross-embodiment learning and enable generalization.

rear right leg further back and maintains a stable limping gait.

C. Understanding Learned Embodiment Representations

To gain insight into the internal representations learned by our policy, we performed a t-SNE [95] analysis on the action latent vectors $\bar{z}_{\text{action}}$ produced by URMA for each embodiment. Figure 6 shows that the learned representations naturally cluster according to the robot morphology, clearly distinguishing humanoids, quadrupeds, and hexapods. For all three morphologies, large clusters around the number of knee joints separate most of the latent space, showing the impact of additional joints on the policy. Many finer sub-clusters emerge based on different geometric and kinematic variations for a given number of knee joints. This structured representation indicates that our policy captures meaningful embodiment-specific features that generalize, mostly, within the morphological classes, whereas patterns across classes are less clear. Additional visualizations using PCA [45] and UMAP [66] can be found in Appendix X.

V. CONCLUSION

We present preliminary empirical evidence for embodiment scaling laws through a large-scale study on robot locomotion, using a procedurally generated dataset GENBOT-1K. Our results show that increasing the number of training embodiments improves generalization to unseen ones, with more challenging morphologies benefiting from continued scaling. Scaling across embodiment classes further enhances generalization, while simply increasing data volume on a fixed set of robots yields diminishing returns. We also demonstrate successful sim-to-real transfer of the learned cross-embodiment policy. As robotic platforms grow more diverse, the ability to learn from and generalize across embodiments becomes increasingly critical. We hope this work offers a step toward scalable, general-purpose robotic agents that generalizes across tasks, environments, and embodiments.

ACKNOWLEDGMENTS

This research is funded by the NSF AI-Center TILOS, the Hillbot Embodied AI Fund, the National Science Centre Poland (Weave programme UMO-2021/43/I/ST6/02711), and by the German Science Foundation (DFG) (grant number PE 2315/17-1).

Co-author Hao Su is the CTO for Hillbot and receives income. The terms of this arrangement have been reviewed and approved by the University of California, San Diego, in accordance with its conflict of interest policies.

We thank the German Research Center for AI (DFKI), Research Department: Systems AI for Robot Learning, for lending the Unitree Go2 and Unitree H1 robots.

Finally, we thank Oleg Kaidanov (DFKI, TU Darmstadt) for his continuous help with the real-world robot deployment.

REFERENCES

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Bo Ai, Wei Gao, Vinay, and David Hsu. Deep visual navigation under partial observability. In *2022 International Conference on Robotics and Automation, ICRA 2022, Philadelphia, PA, USA, May 23-27, 2022*, pages 9439–9446. IEEE, 2022. doi: 10.1109/ICRA46639.2022.9811598. URL https://doi.org/10.1109/ICRA46639.2022.9811598.

[3] Bo Ai, Yuchen Wang, Yugin Tan, and Samson Tan. Whodunit? learning to contrast for authorship attribution. In Yulan He, Heng Ji, Yang Liu, Sujian Li, Chia-Hui Chang, Soujanya Poria, Chenghua Lin, Wray L. Buntine, Maria Liakata, Hanqi Yan, Zonghan Yan, Sebastian Ruder, Xiaojun Wan, Miguel Arana-Catania, Zhongyu Wei, Hen-Hsen Huang, Jheng-Long

Wu, Min-Yuh Day, Pengfei Liu, and Ruifeng Xu, editors, *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2022 - Volume 1: Long Papers, Online Only, November 20-23, 2022*, pages 1142–1157. Association for Computational Linguistics, 2022. URL https://aclanthology.org/2022.aacl-main.84.

[4] Bo Ai, Zhanxin Wu, and David Hsu. Invariance is key to generalization: Examining the role of representation in sim-to-real transfer for visual navigation. In Marcelo H. Ang Jr and Oussama Khatib, editors, *Experimental Robotics*, pages 69–80, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-63596-0.

[5] Takaaki Azakami, Hiroshi Kera, and Kazuhiko Kawamoto. Adversarial body shape search for legged robots. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 682–687. IEEE, 2022.

[6] Teymur Azayev and Karel Zimmerman. Blind Hexapod Locomotion in Complex Terrain with Gait Adaptation Using Deep Reinforcement Learning and Classification. *J Intell Robot Syst*, 2020.

[7] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith LLontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: an open foundation model for generalist humanoid robots. *CoRR*, abs/2503.14734, 2025. doi: 10.48550/ARXIV.2503.14734. URL https://doi.org/10.48550/arXiv.2503.14734.

[8] Nico Bohlinger, Grzegorz Czechmanowski, Maciej Krupka, Piotr Kicki, Krzysztof Walas, Jan Peters, and Davide Tateo. One policy to run them all: an end-to-end learning approach to multi-embodiment locomotion. *Conference on Robot Learning*, 2024.

[9] Nico Bohlinger, Jonathan Kinzel, Daniel Palenicek, Lukasz Antczak, and Jan Peters. Gait in eight: Efficient on-robot learning for omnidirectional quadruped locomotion. *arXiv preprint arXiv:2503.08375*, 2025.

[10] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong T. Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: robotics transformer for real-world control at scale. In Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu, editors, *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. doi: 10.15607/RSS.2023.XIX.025. URL https://doi.org/10.15607/RSS.2023.XIX.025.

[11] Ken Caluwaerts, Atil Iscen, J Chase Kew, Wenhao Yu, Tingnan Zhang, Daniel Freeman, Kuang-Huei Lee, Lisa Lee, Stefano Saliceti, Vincent Zhuang, et al. Barkour: Benchmarking animal-level agility with quadruped robots. *arXiv preprint arXiv:2305.14654*, 2023.

[12] Luigi Campanaro, Siddhant Gangapurwala, Wolfgang Merkt, and Ioannis Havoutis. Learning and deploying robust locomotion policies with minimal dynamics randomization. *arXiv preprint arXiv:2209.12878*, 2022.

[13] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9630–9640. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00951. URL https://doi.org/10.1109/ICCV48922.2021.00951.

[14] Elliot Chane-Sane, Joseph Amigo, Thomas Flayols, Ludovic Righetti, and Nicolas Mansard. Soloparkour: Constrained reinforcement learning for visual locomotion from privileged experience. In *8th Annual Conference on Robot Learning*, 2024.

[15] Yu Chen, Yingfeng Chen, Zhipeng Hu, Tianpei Yang, Changjie Fan, Yang Yu, and Jianye Hao. Learning action-transferable policy with action embedding. *arXiv preprint arXiv:1909.02291*, 2019.

[16] Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. Extreme parkour with legged robots. In *RoboLetics: Workshop on Robot Learning in Athletics@ CoRL 2023*, 2023.

[17] Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive whole-body control for humanoid robots. In Dana Kulic, Gentiane Venture, Kostas E. Bekris, and Enrique Coronado, editors, *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, 2024. doi: 10.15607/RSS.2024.XX.107. URL https://doi.org/10.15607/RSS.2024.XX.107.

[18] Jia-Ruei Chiu, Yu-Chih Huang, Hui-Ching Chen, Kuan-Yu Tseng, and Pei-Chun Lin. Development of a Running Hexapod Robot with Differentiated Front and Hind Leg Morphology and Functionality. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3710–3717, Las Vegas, NV, USA,

October 2020. IEEE. doi: 10.1109/iros45743.2020.9340811. URL https://ieeexplore.ieee.org/document/9340811/.

[19] Suyoung Choi, Gwanghyeon Ji, Jeongsoo Park, Hyeongjun Kim, Juhyeok Mun, Jeong Hyun Lee, and Jemin Hwangbo. Learning quadrupedal locomotion on deformable terrain. *Science Robotics*, 8(74):eade2256, 2023.

[20] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022. URL https://arxiv.org/abs/2204.02311.

[21] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

[22] Ria Doshi, Homer Rich Walke, Oier Mees, Sudeep Dasari, and Sergey Levine. Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation. In *8th Annual Conference on Robot Learning*, 2024.

[23] Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. In Kris Hauser, Dylan A. Shell, and Shoudong Huang, editors, *Robotics: Science and Systems XVIII, New York City, NY, USA, June 27 - July 1, 2022*, 2022. doi: 10.15607/RSS.2022.XVIII.063. URL https://doi.org/10.15607/RSS.2022.XVIII.063.

[24] Ainaz Eftekhar, Luca Weihs, Rose Hendrix, Ege Caglar, Jordi Salvador, Alvaro Herrasti, Winson Han, Eli VanderBil, Aniruddha Kembhavi, Ali Farhadi, et al. The one ring: a robotic indoor navigation generalist. *arXiv preprint arXiv:2412.14401*, 2024.

[25] Haoshu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11441–11450. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.01146. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Fang_GraspNet-1Billion_A_Large-Scale_Benchmark_for_General_Object_Grasping_CVPR_2020_paper.html.

[26] Haoshu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. RH20T: A comprehensive robotic dataset for learning diverse skills in one-shot. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pages 653–660. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10611615. URL https://doi.org/10.1109/ICRA57147.2024.10611615.

[27] Kuan Fang, Patrick Yin, Ashvin Nair, Homer Walke, Gengchen Yan, and Sergey Levine. Generalization with lossy affordances: Leveraging broad offline data for learning visuomotor tasks. In Karen Liu, Dana Kulic, and Jeffrey Ichnowski, editors, *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, volume 205 of *Proceedings of Machine Learning Research*, pages 106–117. PMLR, 2022. URL https://proceedings.mlr.press/v205/fang23a.html.

[28] Gilbert Feng, Hongbo Zhang, Zhongyu Li, Xue Bin Peng, Bhuvan Basireddy, Linzhu Yue, Zhitao Song, Lizhi Yang, Yunhui Liu, Koushil Sreenath, and Sergey Levine. Genloco: Generalized locomotion controllers for quadrupedal robots. In Karen Liu, Dana Kulic, and Jeffrey Ichnowski, editors, *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, volume 205 of *Proceedings of Machine Learning Research*, pages 1893–1903. PMLR, 2022. URL https://proceedings.mlr.press/v205/feng23a.html.

[29] Hiroki Furuta, Yusuke Iwasawa, Yutaka Matsuo, and Shixiang Shane Gu. A system for morphology-task generalization via unified representation and behavior distillation. In *The Eleventh International Conference on Learning Representations*, 2022.

[30] Qiyue Gao, Xinyu Pi, Kevin Liu, Junrong Chen, Ruolan Yang, Xinqi Huang, Xinyu Fang, Lu Sun, Gautham Kishore, Bo Ai, Stone Tao, Mengyang Liu, Jiaxi Yang, Chao-Jung Lai, Chuanyang Jin, Jiannan Xiang, Benhao Huang, David Danks, Hao Su, Tianmin Shu, Ziqiao Ma, Lianhui Qin, and Zhiting Hu. Do vision-language models have internal world models? towards an atomic evaluation. In *ICLR 2025 Workshop on World Models: Understanding, Modelling and Scaling*, 2025. URL https://openreview.net/forum?id=tpPv3ayoqo.

[31] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.295. URL https://aclanthology.org/2021.acl-long.295/.

[32] Wei Gao, Bo Ai, Joel Loo, Vinay, and David Hsu. Intentionnet: Map-lite visual navigation at the kilometre scale, 2024. URL https://arxiv.org/abs/2407.03122.

[33] Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In Dana Kulic, Gentiane Venture, Kostas E. Bekris, and Enrique Coronado, editors, *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, 2024. doi: 10.15607/RSS.2024.XX.090. URL https://doi.org/10.15607/RSS.2024.XX.090.

[34] Agrim Gupta, Linxi Fan, Surya Ganguli, and Li Fei-Fei. Metamorph: Learning universal controllers with transformers. *arXiv preprint arXiv:2203.11931*, 2022.

[35] Haitao Yu, Wei Guo, Jing Deng, Mantian Li, and Hegao Cai. A CPG-based locomotion control architecture for hexapod robot. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5615–5621, Tokyo, November 2013. IEEE. doi: 10.1109/iros.2013.6697170. URL http://ieeexplore.ieee.org/document/6697170/.

[36] Christopher Hazard, Nancy Pollard, and Stelian Coros. Automated design of robotic hands for in-hand manipulation tasks. *International Journal of Humanoid Robotics*, 17(01):1950029, 2020.

[37] Tairan He, Chong Zhang, Wenli Xiao, Guanqi He, Changliu Liu, and Guanya Shi. Agile but safe: Learning collision-free high-speed legged locomotion. In Dana Kulic, Gentiane Venture, Kostas E. Bekris, and Enrique Coronado, editors, *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, 2024. doi: 10.15607/RSS.2024.XX.059. URL https://doi.org/10.15607/RSS.2024.XX.059.

[38] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *CoRR*, abs/2203.15556, 2022. doi: 10.48550/ARXIV.2203.15556. URL https://doi.org/10.48550/arXiv.2203.15556.

[39] Yang Hu and Giovanni Montana. Skill transfer in deep reinforcement learning under morphological heterogeneity. *arXiv preprint arXiv:1908.05265*, 2019.

[40] Wenlong Huang, Igor Mordatch, and Deepak Pathak. One policy to control them all: Shared modular policies for agent-agnostic control. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4455–4464. PMLR, 2020. URL http://proceedings.mlr.press/v119/huang20d.html.

[41] Physical Intelligence, Kevin Black, Noah Brown, James

Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

[42] Fabian Jenelten, Junzhe He, Farbod Farshidian, and Marco Hutter. Dtc: Deep tracking control–a unifying approach to model-based planning and reinforcement-learning for versatile and robust locomotion. *arXiv preprint arXiv:2309.15462*, 2023.

[43] Mazeyu Ji, Xuanbin Peng, Fangchen Liu, Jialong Li, Ge Yang, Xuxin Cheng, and Xiaolong Wang. Exbody2: Advanced expressive humanoid whole-body control. *CoRR*, abs/2412.13196, 2024. doi: 10.48550/ARXIV.2412.13196. URL https://doi.org/10.48550/arXiv.2412.13196.

[44] Zhiwei Jia, Xuanlin Li, Zhan Ling, Shuang Liu, Yiran Wu, and Hao Su. Improving policy optimization with generalist-specialist learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 10104–10119. PMLR, 2022. URL https://proceedings.mlr.press/v162/jia22a.html.

[45] Ian Jolliffe. *Principal component analysis*. Springer Verlag, New York, 2002.

[46] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020. URL https://arxiv.org/abs/2001.08361.

[47] Mohammadreza Kasaei, Miguel Abreu, Nuno Lau, Artur Pereira, and Luis Paulo Reis. A cpg-based agile and versatile locomotion framework using proximal symmetry loss. *arXiv preprint arXiv:2103.00928*, 2021.

[48] Elia Kaufmann, Leonard Bauersfeld, Antonio Loquercio, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Champion-level drone racing using deep reinforcement learning. *Nature*, 620(7976):982–987, 2023.

[49] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R. Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J. Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A large-scale in-the-wild robot manipulation dataset. In Dana Kulic, Gentiane Venture, Kostas E. Bekris, and Enrique Coronado, editors, *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, 2024. doi: 10.15607/RSS.2024.XX.120. URL https://doi.org/10.15607/RSS.2024.XX.120.

[50] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3992–4003, 2023. doi: 10.1109/ICCV51070.2023.00371.

[51] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. *Robotics: Science and Systems XVII*, 2021.

[52] Ashish Kumar, Zhongyu Li, Jun Zeng, Deepak Pathak, Koushil Sreenath, and Jitendra Malik. Adapting rapid motor adaptation for bipedal robots. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1161–1168. IEEE, 2022.

[53] Aviral Kumar, Anikait Singh, Frederik D. Ebert, Mitsuhiko Nakamoto, Yanlai Yang, Chelsea Finn, and Sergey Levine. Pre-training for robots: Offline RL enables learning new tasks in a handful of trials. In Kostas E. Bekris, Kris Hauser, Sylvia L. Herbert, and Jingjin Yu, editors, *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. doi: 10.15607/RSS.2023.XIX.019. URL https://doi.org/10.15607/RSS.2023.XIX.019.

[54] Jacob Levy, Tyler Westenbroek, and David Fridovich-Keil. Learning to walk from three minutes of real-world data with semi-structured dynamics models. In

8th Annual Conference on Robot Learning, 2024.

[55] Qiayuan Liao, Bike Zhang, Xuanyu Huang, Xiaoyu Huang, Zhongyu Li, and Koushil Sreenath. Berkeley humanoid: A research platform for learning-based control. *arXiv preprint arXiv:2407.21781*, 2024.

[56] Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. *CoRR*, abs/2410.18647, 2024. doi: 10.48550/ARXIV.2410.18647. URL https://doi.org/10.48550/arXiv.2410.18647.

[57] Minghuan Liu, Zixuan Chen, Xuxin Cheng, Yandong Ji, Ri-Zhao Qiu, Ruihan Yang, and Xiaolong Wang. Visual whole-body control for legged loco-manipulation. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard, editors, *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, volume 270 of *Proceedings of Machine Learning Research*, pages 234–257. PMLR, 2024. URL https://proceedings.mlr.press/v270/liu25b.html.

[58] Xingyu Liu, Deepak Pathak, and Ding Zhao. Meta-evolve: Continuous robot evolution for one-to-many policy transfer. In *The Twelfth International Conference on Learning Representations*, 2024.

[59] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

[60] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[61] Dhruv Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part II*, volume 11206 of *Lecture Notes in Computer Science*, pages 185–201. Springer, 2018. doi: 10.1007/978-3-030-01216-8_12. URL https://doi.org/10.1007/978-3-030-01216-8_12.

[62] Angelo Maravita and Atsushi Iriki. Tools for the body (schema). *Trends in Cognitive Sciences*, 8(2):79–86, 2004. ISSN 1364-6613. doi: https://doi.org/10.1016/j.tics.2003.12.008. URL https://www.sciencedirect.com/science/article/pii/S1364661303003450.

[63] Gabriel Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid locomotion via reinforcement learning. In *Robotics: Science and Systems*, 2022.

[64] Gabriel B Margolis and Pulkit Agrawal. Walk these ways: Tuning robot control for generalization with multiplicity of behavior. In *Conference on Robot Learning*, pages 22–31. PMLR, 2023.

[65] Gabriel B. Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid locomotion via reinforcement learning. *Int. J. Robotics Res.*, 43(4):572–587, 2024. doi: 10.1177/02783649231224053. URL https://doi.org/10.1177/02783649231224053.

[66] Leland McInnes and John Healy. UMAP: uniform manifold approximation and projection for dimension reduction. *CoRR*, abs/1802.03426, 2018. URL http://arxiv.org/abs/1802.03426.

[67] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022.

[68] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023. doi: 10.1109/LRA.2023.3270034.

[69] R Nezafat, R Shadmehr, and H H Holcomb. Long-term adaptation to dynamics of reaching movements: a PET study. *Experimental Brain Research*, 140(1):66–76, September 2001.

[70] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alexander Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Paul Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guanzhi Wang, Hao Su, Haoshu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I. Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie

Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi Jim Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J. Joshi, Niko Sünderhauf, Ning Liu, Norman Di Palo, Nur Muhammad (Mahi) Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R. Sanketi, Patrick Tree Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham D. Sonawani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Liangwei Xu, Xuanlin Li, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and RT-X models : Open x-embodiment collaboration. In *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, pages 6892–6903. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10611477. URL https://doi.org/10.1109/ICRA57147.2024.10611477.

[71] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

[72] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

[73] Wenjuan Ouyang, Haozhen Chi, Jiangnan Pang, Wenyu Liang, and Qinyuan Ren. Adaptive locomotion control of a hexapod robot via bio-inspired learning. *Frontiers Neurorobotics*, 15:627157, 2021. doi: 10.3389/FNBOT.2021.627157. URL https://doi.org/10.3389/fnbot.2021.627157.

[74] Austin Patel and Shuran Song. Get-zero: Graph embodiment transformer for zero-shot embodiment generalization. *CoRR*, abs/2407.15002, 2024. doi: 10.48550/ARXIV.2407.15002. URL https://doi.org/10.48550/arXiv.2407.15002.

[75] Austin Patel and Shuran Song. Get-zero: Graph embodiment transformer for zero-shot embodiment generalization. *arXiv preprint arXiv:2407.15002*, 2024.

[76] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018.

[77] Tomson Qu, Dichen Li, Avideh Zakhor, Wenhao Yu, and Tingnan Zhang. Versatile locomotion skills for hexapod robots. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6885–6892, 2024. doi: 10.1109/IROS58592.2024.10801714.

[78] Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Real-world humanoid locomotion with reinforcement learn-

ing. *arXiv:2303.03381*, 2023.

[79] Chang Rajani, Karol Arndt, David Blanco-Mulero, Kevin Sebastian Luck, and Ville Kyrki. Co-imitation: learning design and behaviour by imitation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6200–6208, 2023.

[80] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100. PMLR, 2022.

[81] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[82] Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation. In Dana Kulic, Gentiane Venture, Kostas E. Bekris, and Enrique Coronado, editors, *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, 2024. doi: 10.15607/RSS.2024.XX.061. URL https://doi.org/10.15607/RSS.2024.XX.061.

[83] Milad Shafiee, Guillaume Bellegarda, and Auke 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.

[84] Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. Gnm: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7226–7233. IEEE, 2023.

[85] Haochen Shi, Weizhuo Wang, Shuran Song, and C. Karen Liu. Toddlerbot: Open-source ml-compatible humanoid platform for loco-manipulation, 2025. URL https://arxiv.org/abs/2502.00893.

[86] Jonah Siekmann, Kevin Green, John Warila, Alan Fern, and Jonathan Hurst. Blind bipedal stair traversal via sim-to-real reinforcement learning. In *Robotics: Science and Systems*, 2021.

[87] Laura Smith, Ilya Kostrikov, and Sergey Levine. A walk in the park: Learning to walk in 20 minutes with model-free reinforcement learning. *arXiv preprint arXiv:2208.07860*, 2022.

[88] Laura Smith, Yunhao Cao, and Sergey Levine. Grow your limits: Continuous improvement with real-world rl for robotic locomotion. *arXiv preprint arXiv:2310.17634*, 2023.

[89] Wenxuan Song, Han Zhao, Pengxiang Ding, Can Cui, Shangke Lyu, Yaning Fan, and Donglin Wang. Germ: A generalist robotic model with mixture-of-experts for quadruped robot. *arXiv preprint arXiv:2403.13358*, 2024.

[90] M. Stasica, A. Bick, N. Bohlinger, O. Mohseni, J. Fritzsche, C. Hübler, J. Peters, and A. Seyfarth. Bridge the gap: Enhancing quadruped locomotion with vertical ground perturbations. In *Under review*, 2025. URL https://www.ias.informatik.tu-darmstadt.de/uploads/Team/NicoBohlinger/bridge_the_gap.pdf.

[91] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 843–852. IEEE Computer Society, 2017. doi: 10.1109/ICCV.2017.97. URL https://doi.org/10.1109/ICCV.2017.97.

[92] Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. In Hadas Kress-Gazit, Siddhartha S. Srinivasa, Tom Howard, and Nikolay Atanasov, editors, *Robotics: Science and Systems XIV, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, June 26-30, 2018*, 2018. doi: 10.15607/RSS.2018.XIV.010. URL http://www.roboticsproceedings.org/rss14/p10.html.

[93] Tongxuan Tian, Haoyang Li, Bo Ai, Xiaodi Yuan, Zhiao Huang, and Hao Su. Diffusion dynamics models with generative state estimation for cloth manipulation. *CoRR*, abs/2503.11999, 2025. doi: 10.48550/ARXIV.2503.11999. URL https://doi.org/10.48550/arXiv.2503.11999.

[94] Brandon Trabucco, Mariano Phielipp, and Glen Berseth. Anymorph: Learning transferable polices by inferring agent morphology. In *International Conference on Machine Learning*, pages 21677–21691. PMLR, 2022.

[95] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

[96] Homer Rich Walke, Kevin Black, Tony Z. Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, Abraham Lee, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata V2: A dataset for robot learning at scale. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine Learning Research*, pages 1723–1736. PMLR, 2023. URL https://proceedings.mlr.press/v229/walke23a.html.

[97] Weikang Wan, Haoran Geng, Yun Liu, Zikang Shan, Yaodong Yang, Li Yi, and He Wang. Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 3868–3879. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00360. URL https://doi.org/10.1109/ICCV51070.2023.00360.

[98] Tingwu Wang, Renjie Liao, Jimmy Ba, and Sanja Fidler. Nervenet: Learning structured policy with graph neural

networks. In *International Conference on Learning Representations*, 2018.

[99] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 17868–17879. IEEE, 2024. doi: 10.1109/CVPR52733.2024.01692. URL https://doi.org/10.1109/CVPR52733.2024.01692.

[100] DA Winter. *Biomechanics and Motor Control of Human Movement*, chapter 10, pages 250–280. John Wiley & Sons, New York, 2nd edition, 1990.

[101] Zhanxin Wu, Bo Ai, and David Hsu. Integrating common sense and planning with large language models for room tidying. In *RSS 2023 Workshop on Learning for Task and Motion Planning*, 2023. URL https://openreview.net/forum?id=vuSI9mhDaBZ.

[102] Ruihan Yang, Minghao Zhang, Nicklas Hansen, Huazhe Xu, and Xiaolong Wang. Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nhnJ3oo6AB.

[103] Zixian Zang, Maxime Kawawa-Beaudan, Wenhao Yu, Tingnan Zhang, and Avideh Zakhor. Perceptive Hexapod Legged Locomotion for Climbing Joist Environments. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2738–2745, Detroit, MI, USA, October 2023. IEEE. ISBN 978-1-66549-190-7. doi: 10.1109/IROS55552.2023.10341957. URL https://ieeexplore.ieee.org/document/10341957/.

[104] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 1204–1213. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01179. URL https://doi.org/10.1109/CVPR52688.2022.01179.

[105] He Zhang, Yubin Liu, Jie Zhao, Jie Chen, and Jihong Yan. Development of a Bionic Hexapod Robot for Walking on Unstructured Terrain. *Journal of Bionic Engineering*, 11(2):176–187, June 2014. ISSN 2543-2141. doi: 10.1016/S1672-6529(14)60041-X. URL https://doi.org/10.1016/S1672-6529(14)60041-X.

[106] Allan Zhao, Jie Xu, Mina Konaković-Luković, Josephine Hughes, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. Robogrammar: graph grammar for terrain-optimized robot design. *ACM Transactions on Graphics (TOG)*, 39(6):1–16, 2020.

[107] Ruiqi Zhu, Tianhong Dai, and Oya Celiktutan. Cross domain policy transfer with effect cycle-consistency. In *2024 IEEE International Conference on Robotics and Automation*. IEEE Explore, 2024.

[108] Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher Atkeson, Sören Schwertfeger, Chelsea Finn, and Hang Zhao. Robot parkour learning. In *Conference on Robot Learning (CoRL)*, 2023.

[109] Ziwen Zhuang, Shenzhe Yao, and Hang Zhao. Humanoid parkour learning. *arXiv preprint arXiv:2406.10759*, 2024.

[110] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong T. Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, volume 229 of *Proceedings of Machine Learning Research*, pages 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

Appendix

VI. Expert Training

A. Observation and Action Space

The observation space of the expert policies includes the joint angles, joint velocities, previous actions, trunk angular velocities, gravity vector and the command velocities. The observation space of the critics includes the same observations as for the policies, but also includes privileged information: the trunk linear velocity, trunk height over the ground, feet contact states and feet air times.

The policies control the robots at 50 Hz with a PD controller, where the target joint angles are generated by scaling the action of the policy and adding it to the nominal joint configuration of the robot: $q_{\text{target}} = q_{\text{nominal}} + \sigma \cdot a$. We define the nominal joint configuration as a standing pose of a robot and use the same configuration for all robots of the same morphology class (see Appendix VII-C). For the action scaling factor σ, we use 0.3 for quadrupeds and hexapods, and 0.75 for humanoids. For the PD controller, we use $K_p = 20$ and $K_d = 0.5$ for quadrupeds, $K_p = 25$ and $K_d = 0.5$ for hexapods, and $K_p = 60$ and $K_d = 2.0$ for humanoids.

B. Domain Randomization

To enable sim-to-real transfer of the trained policies, we add strong domain randomization during training. We use a performance-based curriculum learning approach, where the domain randomization ranges are increased from 0 (or their mean if not zero-centered) to the final values in Table I over the course of training. This curriculum approach allows the policy to learn basic locomotion first in the simplest possible environment before adapting to wider variations. We define a curriculum coefficient from 0 to 1, which is multiplied with the domain randomization ranges (and the reward penalty coefficients). The coefficient of an environment is increased by 0.01 if the policy completed the episode without falling, and the average tracking error of the target x,y velocity is below 0.4 m/s, and the coefficient is reduced by 0.01 otherwise.

Every embodiment in GenBot-1K uses the same domain randomization ranges. The "starting" values (naming scheme in Table I) are sampled uniformly at the start of every episode to randomize the starting state of the robot. The "noise" values are sampled uniformly for every simulation step to add noise to the observations. The values of every other parameter are sampled uniformly every simulation step with a probability of 0.002 (on average every 500 steps / every 10 seconds). Pushes are applied as linear velocities to the trunk of the robot.

C. Reward Function

Table II contains all reward terms and coefficients of the reward function for the expert training of all robots in the GenBot-1K dataset. Joint-based (T6-T12) and feet-based (T14-T17) reward terms are calculated as the mean over every joint and foot, respectively, to account for the varying amounts of joints and feet of the generated embodiments. The coefficients of all penalties are attached to the curriculum coefficient (see Appendix VI-B) and thus linearly increase from 0 to the final values in Table II over the course of training. This makes the training process less sensitive to the precise values of the coefficients.

TABLE I: **Domain randomization configuration.** Domain randomization values and ranges for every randomized parameter during the expert RL training. The values in the table are the maximum values and ranges in the curriculum when reaching the final curriculum coefficient of 1.

Parameter	Value
Max action delay	1
Chance for action delay	0.05
Min & max motor strength	(0.5, 1.5)
Min & max P gain factor	(0.5, 1.5)
Min & max D gain factor	(0.5, 1.5)
Min & max joint position offset	(-0.05, 0.05)
Min & max starting orientation factor	(-0.0625, 0.0625)
Min & max starting joint position factor	(-0.5, 0.5)
Min & max starting joint velocity factor	(-0.5, 0.5)
Min & max starting linear velocity	(-0.5, 0.5)
Min & max starting angular velocity	(-0.5, 0.5)
Joint position noise	0.01
Joint velocity noise	1.5
Angular velocity noise	0.2
Gravity velocity noise	0.05
Joint observation dropout chance	0.05
Min & max static friction	(0.05, 2.0)
Min & max dynamic friction	(0.05, 1.5)
Min & max restitution	(0.0, 1.0)
Min & max added mass	(-2.0, 2.0)
Min & max gravity	(-8.81, 10.81)
Min & max joint friction	(0.0, 0.01)
Min & max joint armature	(0.0, 0.01)
Min & max pushes in x	(-1.0, 1.0)
Min & max pushes in y	(-1.0, 1.0)
Min & max pushes in z	(-1.0, 1.0)

D. PPO Hyperparameters

We use the same PPO hyperparameters for the training of all expert policies, detailed in Table III. Searching for better hyperparameters for every embodiment might lead to increased performance but is impractical when considering training $\sim$1000 embodiments. The chosen hyperparameters are based on common practices in legged locomotion research [8, 80] and preliminary tuning on a small subset of embodiments.

VII. Embodiment Generation

A. Basic Units for Quadruped, Humanoid and Hexapod

Tables IV and V provide the base values for geometry-related and kinematics-related parameters, respectively, for representative links across quadruped, humanoid, and hexapod morphologies. For the humanoid class, we report parameters for the trunk and the left-side lower-body links. For quadruped

TABLE II: **Reward terms for the RL training of embodiment-specific experts.** All reward terms and the corresponding coefficients that compose the reward function for the expert training. While all the coefficients work for all embodiments, for the final experiments, we tweaked four coefficients for the humanoid embodiments to improve the style of the gait: *1 3.0, *2 1.5, *3 43.2, *4 6e-3.

	Term	Equation	Coefficient
T1	Xy velocity tracking	$\exp(-\lvert v_{xy} - c_{xy}\rvert^2/0.25)$	2.0 *1
T2	Yaw velocity tracking	$\exp(-\lvert\omega_{\text{yaw}} - c_{\text{yaw}}\rvert^2/0.25)$	1.0 *2
T3	Z velocity penalty	$-\lvert v_z\rvert^2$	2.0
T4	Pitch-roll velocity penalty	$-\lvert\omega_{\text{pitch, roll}}\rvert^2$	0.05
T5	Pitch-roll position penalty	$-\lvert\theta_{\text{pitch, roll}}\rvert^2$	5.0
T6	Joint nominal differences penalty	$-\lvert q - q^{\text{nominal}}\rvert^2$	14.4 *3
T7	Joint position limits penalty	$-\bar{\mathbb{1}}(0.9q_{\text{min}} < q < 0.9q_{\text{max}})$	120.0
T8	Joint velocity limits penalty	$-\bar{\mathbb{1}}(0.9\dot{q}_{\text{min}} < \dot{q} < 0.9\dot{q}_{\text{max}})$	10.0
T9	Joint accelerations penalty	$-\lvert\ddot{q}\rvert^2$	5e-6
T10	Joint torques penalty	$-\lvert\tau\rvert^2$	2.4e-4
T11	Action rate penalty	$-\lvert a_t - a_{t-1}\rvert^2$	0.12
T12	Action smoothness penalty	$-\lvert a_t - 2a_{t-1} + a_{t-2}\rvert^2$	0.12
T13	Walking height penalty	$-\lvert h - h_{\text{nominal}}\rvert^2$	30.0
T14	Air time penalty	$-\sum_f \mathbb{1}(p_f)(p_f^T - 0.5)$	0.1
T15	Symmetry penalty	$-\sum_f \bar{\mathbb{1}}(p_f^{\text{left}})\bar{\mathbb{1}}(p_f^{\text{right}})$	0.5
T16	Feet y distance penalty	$-\lvert f^{\text{actual}}_{\text{y distance}} - f^{\text{target}}_{\text{y distance}}\rvert^2$	2.0
T17	Feet force penalty	$-\lvert f_{\text{force}}\rvert^2$	8e-3 *4
T18	Self-collision penalty	$-\mathbb{1}_{\text{self-collision}}$	1.0

TABLE III: **PPO hyperparameters for expert policy training.**

Hyperparameter	Value
Batch size	98304
Mini-batch size	24576
# epochs	5
Initial learning rate	0.001
Learning rate schedule	Adaptive with target KL 0.01
Entropy coefficient	0.002
Discount factor	0.99
GAE λ	0.95
Clip range	0.2
Max gradient norm	1.0
Initial action standard deviation	1.0
Clip range action mean	-10.0, 10.0
Policy and critic hidden layers	[512, 256, 128]
Activation function	ELU
# training iterations	17500 (quadruped, hexapod), 42500 (humanoid)

and hexapod classes, we include the front-left leg. The remaining components are either symmetric or peripheral to locomotion (e.g., arms or head for humanoids) and therefore omitted.

The base parameter values are partially inspired by the Unitree Go2 and H1 platforms, offering a degree of realism without exact replication. This design choice is consistent with prior work such as GenLoco [28], which abstracts physical characteristics from real robots to define a diverse yet grounded design space. Robots instantiated with these values correspond to a $1.0\times$ variation setting (i.e., no geometric, kinematic, or topological scaling applied), and serve as the reference point for applying the variation factors listed in Tables IV and V.

To support meaningful evaluation of generalization, these reference robots are excluded from the training set. Every robot in the training set differs from Go2 and H1 by at least one geometric, topological, or kinematic variation, along with additional discrepancies due to loose alignment in parameter values (e.g., each joint in the humanoid closest to H1 differs by a few centimeters, and the overall height differs by approximately 10 cm). This diversity encourages the learned policy

TABLE IV: **Base geometry and mass parameters for representative link types in the embodiment generation pipeline used in GENBOT-1K.** Geometry dimensions are specified according to shape type: Sphere (radius), Cylinder (length, radius), and Box (length, width, height).

Class	Link Name	Geometry Type	Geometry Dimension (m)	Mass (kg)
Humanoid	Pelvis	Sphere	(0.05,)	5.390
	Torso	Box	(0.08, 0.26, 0.18)	17.789
	Hip yaw link	Cylinder	(0.02, 0.01)	2.244
	Hip roll link	Cylinder	(0.01, 0.02)	2.232
	Thigh	Cylinder	(0.2, 0.05)	4.152
	Calf	Cylinder	(0.2, 0.05)	1.721
	Foot	Box	(0.28, 0.03, 0.024)	0.474
Quadruped	Trunk	Box	(0.38, 0.09, 0.11)	6.921
	Hip	Cylinder	(0.04, 0.046)	1.152
	Thigh	Box	(0.21, 0.025, 0.034)	1.152
	Calf	Cylinder	(0.12, 0.013)	0.154
g	Foot	Sphere	(0.022,)	0.040
Hexapod	Trunk	Box	(0.8, 0.5, 0.1)	6.921
	Hip	Sphere	(0.05,)	0.678
	Thigh	Cylinder	(0.22, 0.03)	1.152
	Calf	Cylinder	(0.22, 0.025)	0.154
	Foot	Sphere	(0.03,)	0.040

TABLE V: **Motor and joint properties of the generated embodiments in GENBOT-1K**.

Class	Joint Name	Joint Limits (rad)	Max. Torque (N·m)	Max. Velocity (rad/s)
Humanoid	Torso joint	(-2.35, 2.35)	200	23
	Shoulder pitch joint	(-2.87, 2.87)	40	9
	Shoulder roll joint	(-0.34, 3.11)	40	9
	Shoulder yaw joint	(-1.30, 4.45)	18	20
	Elbow joint	(-1.25, 2.61)	18	20
	Hip yaw/roll joint	(-0.43, 0.43)	200	23
	Hip pitch	(-3.10, 2.50)	200	23
	Knee joint	(-0.26, 2.00)	300	14
	Ankle joint	(-0.87, 0.52)	40	9
Quadruped	Hip pitch joint	(-1.05, 1.05)	23.7	30.1
	Front thigh joint	(-1.57, 3.49)	23.7	30.1
	Rear thigh joint	(-0.52, 4.53)	23.7	30.1
	Knee joint	(-2.72, -0.84)	45.43	15.7
Hexapod	Hip joint	(-1.57, 1.57)	100	30
	Thigh joint	(-1.57, 1.57)	100	30
	Knee joint	(-1.57, 1.57)	100	30

to capture broadly transferable motion patterns. As discussed in Section IV, empirical results suggest that the policy has acquired sufficiently generalizable behaviors to support both cross-embodiment and sim-to-real transfer, which is generally considered highly challenging.

B. Generation Algorithm

We construct each robot embodiment in a tree-like structure by iteratively connecting links using joints, following the URDF specification and the basic units described in Section VII-A. The construction procedure varies slightly across morphologies:

- **Humanoids**: The root node is the pelvis. We first append the torso and hip links, then attach the shoulder and arm links for the upper body, followed by the thigh, calf, and foot links for the lower body.
- **Quadrupeds and hexapods**: The root node is the trunk. We sequentially append the hip links to the trunk, then connect the leg and foot links to form the complete body.

To ensure diversity in the generated embodiments, we intro-

TABLE VI: **Variation parameters across geometry, topology, and kinematics in the embodiment generation algorithm.** The torso link randomization only applicable to the humanoid class.

Variation Type	Parameter Name	Candidate Values
Topology	Number of knee joints	{0, 1, 2, 3}
Geometry	Scaling factor for all link size	{0.8, 1.0, 1.2}
	Scaling factor for thigh link length	{0.4, 0.8, 1.0, 1.2, 1.6}
	Scaling factor for calf link length	{0.4, 0.8, 1.0, 1.2, 1.6}
	Scaling factor for foot link size	{1.0, 2.0}
	Scaling factor for torso link size	{0.4, 0.8, 1.0, 1.2, 1.6}
Kinematics	Scaling factor for knee joint limits	{0.2, 0.6, 1.0}

duce variations in geometry, topology, and kinematics during the construction process, as detailed in Section III. Table VI summarizes the variation parameters and their corresponding candidate values. While most parameters are self-explanatory, we clarify a few specific cases:

- **Number of knee joints**: If a leg is configured with zero knee joints, the calf link is omitted, and the thigh link is directly connected to the foot.
- **Foot link size**: For humanoids, foot links are modeled as boxes and scaled by length; for quadrupeds and hexapods, foot links are modeled as spheres and scaled by radius.
- **Joint limit variation**: Joint limits are varied by uniformly scaling the nominal joint ranges about the nominal joint position, which serves as a fixed point.

C. Nominal Joint Configurations

Nominal joint configurations are used to initialize robot poses during training, contribute to reward terms that discourage deviations too far from these default joint angles, and function as offsets to the actions of the expert and distillation policies. As such, they serve as useful regularizers for learning realistic and efficient gaits. To support scalability across diverse morphologies, we generate nominal configurations by reusing unit values across the generated embodiments. The nominal joint angles used are summarized in Table VII.

TABLE VII: **Nominal joint configurations for generated embodiments in GenBot-1K.** These joint angles are used to initialize robot poses, define regularization rewards, and function as offsets to the policy actions. The values are consistent across symmetric limbs.

Class	Joint Name	Joint Angle (rad)
Humanoid	Torso	0.0
	Shoulder (Left/Right, pitch/roll/yaw)	0.0
	Elbow (Left/Right)	0.0
	Hip pitch (Left/Right)	-0.4
	Hip roll/yaw (Left/Right)	0.0
	Knee (Left/Right)	0.8
	Ankle (Left/Right)	-0.4
Quadruped	Hip (Front/Rear, Left/Right)	±0.1
	Thigh (Front, Left/Right)	0.8
	Thigh (Rear, Left/Right)	1.0
	Knee (Front/Rear, Left/Right)	-1.5
	Additional knee joints (if any)	0.0
Hexapod	Hip (Front/Middle/Rear, Left/Right)	0.0
	Thigh (Front/Middle/Rear, Left/Right)	0.79
	Knee (Front/Middle/Rear, Left/Right)	0.79
	Additional knee joints (if any)	0.0

VIII. Cross-Embodiment Distillation

A. Expert Data Collection

For every embodiment, we run the expert RL policy for 600 simulation steps using 4096 parallel environments. This results in a total of 1,985,740,800 data samples across all training embodiments. Note that the episode length during the expert training is 1000 simulation steps (equivalent to 20 physical seconds), thus, the collected data only covers the first half of the episode. Using the full length may provide more time-correlated data, which we did not analyze due to time constraints. The final dataset needs around 5 TB of storage using the `h5py` format without additional compression.

B. URMA Architecture Details

The observation space of the URMA policy is split into two parts: joint-specific observations o_j and general observations o_g. The joint-specific observations o_j include the joint angle, joint velocity, previous action of the joint (shape: ($j(e)$, 3)). The general observations o_g include the trunk linear velocity, gravity vector, command velocities, PD gains, action scaling factor, total mass of the robot, robot dimensions, number of joints and feet size (shape: (20,)).

The description vectors d_j of the joints include the relative carthesian position of the joint in the nominal configuration, joint rotation axis, joint nominal angle, maximum joint torque, maximum joint velocity, joint position limits, p-gain, d-gain and action scaling factor, robot mass and dimensions (shape: ($j(e)$, 18)).

We build on the original URMA neural network architecture, as shown in Figure 7, from Bohlinger et al. [8] with the following modifications:

- We use multi-headed attention for the encoding of the joint observations and descriptions to increase the expressiveness of the policy. All our experiments use 3 attention heads.
- We remove the feet-specific attention encoder as not all robots in the real world have foot-specific sensors, like pressure sensors.

TABLE VIII: **Train-test splits of GENBOT-1K.** Each index refers to one unique embodiment in each embodiment class. The training set is simply the complement of the test set and thus omitted.

Class	Test Set
Humanoid	[0, 7, 12, 20, 31, 32, 37, 41, 46, 47, 48, 50, 51, 55, 63, 71, 72, 75, 97, 104, 111, 113, 122, 124, 128, 132, 133, 144, 149, 154, 155, 158, 161, 163, 166, 169, 170, 181, 183, 197, 204, 207, 215, 222, 226, 229, 241, 244, 248, 250, 252, 258, 260, 261, 266, 272, 276, 278, 280, 282, 286, 290, 298, 308, 312, 313, 316, 320, 327, 342]
Quadruped	[0, 7, 8, 20, 31, 32, 37, 41, 46, 47, 48, 50, 51, 55, 71, 72, 75, 97, 104, 111, 113, 122, 124, 128, 132, 133, 144, 149, 154, 155, 158, 161, 163, 166, 169, 170, 181, 183, 197, 204, 207, 215, 222, 226, 229, 241, 244, 248, 250, 252, 258, 260, 261, 266, 272, 278, 280, 282, 286, 290, 298, 308, 312, 313, 316, 320, 327]
Hexapod	[0, 7, 8, 20, 31, 32, 37, 41, 46, 47, 48, 50, 51, 55, 71, 72, 75, 97, 104, 111, 113, 122, 124, 128, 132, 133, 144, 149, 154, 155, 158, 161, 163, 166, 169, 170, 181, 183, 197, 204, 207, 215, 222, 226, 229, 241, 244, 248, 250, 252, 258, 260, 261, 266, 272, 278, 280, 282, 286, 290, 298, 308, 312, 313, 316, 320, 327]

TABLE IX: **Statistics of train-test splits of GENBOT-1K.** The splits have an approximately balanced distribution over different categories.

Class	Total Number	Train Set (80%)	Test Set (20%)
Humanoid	348	278	70
Quadruped	332	265	67
Hexapod	332	265	67
Total	1012	808	204

- We directly use the output from the action decoder μ_ν as the action of the policy, instead of using an additional head to produce a standard deviation and sampling from a Gaussian distribution, as we train the policy with imitation learning instead of RL.
- We add another encoding layer to the general observations o_g to project them into a higher dimensional latent space before concatenating them with all the joint latent vectors from the attention heads.
- We use wider feedforward layers (2× the hidden dimensions) throughout the network.

The resulting model has 2.1 million parameters. Overall, it is a compact network with strong inductive biases that leverage the compositional structure of robots.

When applying the actions of the policy to the robots, we use the same PD controllers with the same nominal joint configurations and action scaling factors as in the expert training (see Appendix VI-A).

C. Train-Test Set Splits

We split GENBOT-1K into a training set (80%) and a test set (20%) using a deterministic pseudo-random sampler with a fixed seed, ensuring full reproducibility. The same sampling procedure is applied independently to each morphology class, except for quadrupeds and hexapods, which share identical splits due to matched dataset sizes. Detailed test indices are listed in Table VIII, and summary statistics for each category are shown in Table IX.

D. Training Details

We designed an efficient training pipeline that balances disk I/O, CPU preprocessing, GPU utilization, and RAM usage. Instead of loading every minibatch directly from disk, we first load a fixed number of data slices, each containing a small subset of steps from multiple robot embodiments, into an in-memory buffer. Each slice consists of 100 trajectories with 128 steps per trajectory. Once the buffer is filled, minibatches are sampled uniformly at random, without replacement, until every sample has been seen a fixed number of times. This strategy reduces disk access overhead, improves memory locality, and maintains sample diversity throughout training, though it may introduce local overfitting and biased gradient estimates.

Because data from different robots have varied observation and action spaces, we load them separately and use gradient accumulation to reduce bias in the gradient estimation. Specifically, gradients are accumulated across multiple minibatches before each optimizer step, helping to balance contributions across robot embodiments. While effective, this approach still suffers from local gradient bias. A more principled solution would involve zero-padding to form large, uniform batches across robots, but implementing this would require architectural and pipeline-level changes, which we did not pursue due to time constraints. In theory, this could lead to smoother optimization and potentially better final performance.

To ensure numerical stability, we apply gradient clipping with a maximum norm of 5. We use the AdamW optimizer [60] with β_1=0.9, β_2=0.999, and a cosine-annealed weight decay schedule that decays from 3×10^{-4} to 0 over the course of training [59]. The key hyperparameters for distillation are summarized in Table X.

Our pipeline requires 128 GB of RAM to maintain the in-memory buffer. Due to the small size of the URMA policy, training can be efficiently performed on a single GPU (e.g., NVIDIA RTX 4090 or H100). We did not observe significant gains in convergence from increasing batch size, possibly due to the structured nature and potential bottlenecks in the model architecture. Further investigation into the scaling behavior of the training dynamics is left for future work.

IX. ADDITIONAL DETAILS ON REAL-WORLD DEPLOYMENT

A. Hardware Setup

We evaluated our distilled URMA policy zero-shot on two real-world platforms: the Unitree Go2 quadruped and the Unitree H1 humanoid. For each robot, we used its URDF to produce the embodiment description vectors d_j. Before

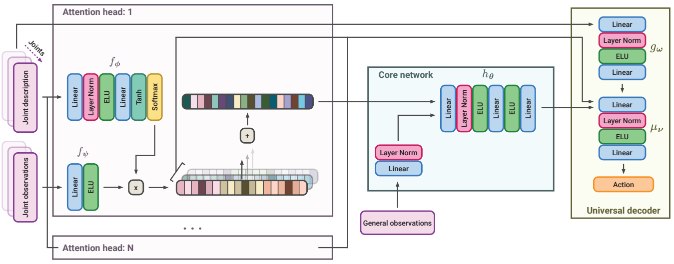


Fig. 7: **URMA with multi-head attention.** We extend the original URMA module [8] with multiple attention heads, each aggregating information from joint observations using distinct attention distributions. This design enables the model to capture multi-modal dependencies and improves its capacity to scale across diverse embodiments.

TABLE X: **Hyperparameters of the distillation pipeline.**

Hyperparameter	Value
# training samples per embodiment	500 × 4096
Validation set size	100 × 4096
Batch size	64
Gradient accumulation steps	8
Gradient clipping threshold	5
Data slice size	100 × 128
Max slices in buffer	1024
Buffer repeat factor	3
Optimizer	AdamW [60]
AdamW betas	(0.9, 0.999)
Weight decay schedule	$3 \times 10^{-4} \rightarrow 0$ (cosine)
Learning rate schedule	Cosine annealing [59]
# epochs	80

deployment, the policy was converted to the ONNX format to load it in JAX and guarantee maximum inference speed. The policy inference ran on a Ubuntu 22.04 laptop (Ryzen 9 CPU), interfaced to the robot over a dedicated Ethernet connection. We ran the control loop at the same 50 Hz and with the same PD gains as in simulation, and sent the target joint angles to the robot's internal controller. We limited the commanded x-y-yaw velocity to 0.8 m/s for the Go2 and 0.5 m/s for the H1, to ensure the robot's stability and safety during the experiments.

B. Implementing Joint Limit Variations

To probe robustness under kinematic constraints, we impose an artificial knee-joint range limited to 20 % of its nominal span. In simulation, one can enforce such limits by directly clamping joint angles within the physics engine; in hardware, however, neither the robot's encoders nor its embedded PD controller can be modified. Consequently, we introduce a software-level joint-limit layer into the control loop in order to restrict the target joint angle for affected knee joints to the new limits. At each control step, the policy's commanded knee angle is constrained to the prescribed ±20 % bounds. Instead, we implemented a software-based solution that restricts the target joint angle for affected knee joints to the new limits. To counteract any excursions driven by external disturbances, we implement an active rejection mechanism: whenever the measured knee angle violates the software limits, we (1) project the commanded target onto the nearest permissible bound and (2) elevate the proportional and derivative gains to $K_p = 60$ and $K_d = 1$, respectively, until the joint re-enters the safe region. This procedure enforces a soft joint-limit constraint exclusively in software—without altering hardware or contravening physical laws—while delivering high-gain corrective action against environmental perturbations.

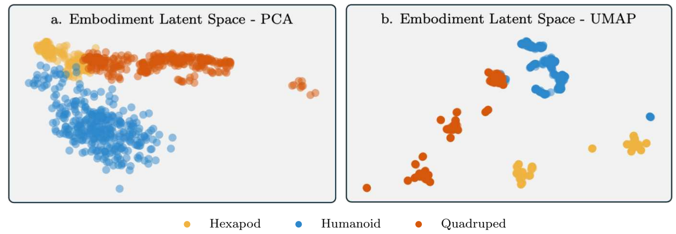


Fig. 8: **Additional visualizations of the learned embodiment embeddings.** Principal Component Analysis (PCA) (a.) and Uniform Manifold Approximation and Projection (UMAP) (b.) of the embodiment latent space (i.e., every point represents one robot, aggregated from all of its joint description vectors).

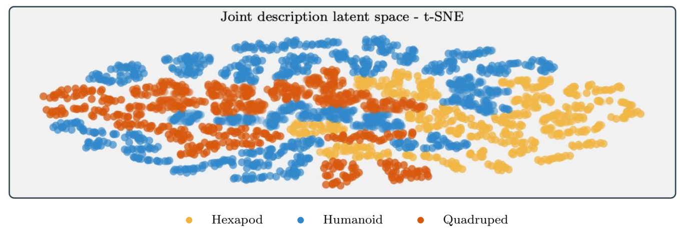


Fig. 9: **Additional visualizations of the learned joint description embeddings.** t-SNE visualization of the joint description latent space of all joints from all embodiments in the GENBOT-1K dataset (i.e., every point represents one joint of a robot).

X. Additional Latent Space Analysis

In addition to the t-SNE analysis, we also apply PCA [45] and UMAP [66] on the action latent vectors $\bar{z}_{\text{action}}$ in Figure 8. Both PCA and UMAP projections reveal clear grouping according to the morphology class, with humanoid, quadruped, and hexapod embeddings forming distinct clusters. Compared to the t-SNE analysis, clusters about the topological, geometric, and kinematic variations are less pronounced and appear to be more cramped.

Furthermore, we show in Figure 9 the t-SNE analysis of the learned joint description latent space $f_\phi(d_j)$ for all joints from all embodiments in the GENBOT-1K dataset. Although the three morphologies still define the rough structure of this latent space, the learned embeddings for the joint descriptions seem to be much more entangled across the three morphology classes.