KINE2GO: KINEMATIC DATASET FOR THE UNITREE GO2 ROBOT WITH DIVERSE GAITS AND MOTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

The recent popularity of robotics, combined with the steadily decreasing cost of robotic hardware, has lowered the entry barrier to robotics research and enabled rapid advancements in the field. One of the primary examples is the Unitree Go2 quadruped robot, which is often used by researchers in the areas of locomotion, navigation, control, and others. Many researchers use the Go2 robot in combination with techniques like imitation learning, reinforcement learning, and behavioral cloning to allow machine learning systems to take full control of the robot. At the same time, many of those techniques require demonstration data consisting of the robot's kinematics information and actions applied to the motors. Obtaining such data is difficult, requires building complex pipelines, and can take significant time. To aid in those kinds of efforts, we present `Kine2Go`, a state-action locomotion dataset for the Unitree Go2 robot comprising 40 motion families, each associated with an RL tracking policy and represented by 20 stochastic rollouts. Our pipeline accepts data from various quadruped morphologies and translates them to a Go2-compatible format. Then we use Reinforcement Learning to train policies following a given motion, and finally we gather data from those policies, which yields perturbed kinematic trajectories with corresponding motor-level actions. To showcase the value of our dataset, we also train a Behavioral Foundation Model (BFM) on it and evaluate its motion-tracking and zero-shot reward-conditioning capabilities.

1 INTRODUCTION

The newest generation of robots can achieve unprecedented speed and agility while moving, even through complex terrain. Most of these feats, however, are achieved through low-level control techniques like PIDs, MPC, etc. Many learning-based locomotion policies are optimized for a particular task, command distribution, or desired behavior. For many real-world robotics applications, it is desirable to produce motions that are smooth, natural, symmetric, and animal-like, which demands separate regularization methods for RL.

Depending on the exact method, regularizing policies can be achieved through many different techniques. In Reinforcement Learning, it is possible to design a reward that will produce a motion with desired properties. However, designing those rewards, often called "reward shaping", is a hard, tedious task requiring a high degree of trial and error. At the same time, RL has a tendency to try to bypass or hack the rewards, which makes the problem even more difficult (Clark and Amodei, 2016).

The most straightforward approach is motion imitation (Peng et al., 2020) or behavioral cloning, where a policy is trained to mimic an expert's trajectories. Those methods tend to yield desirable results; however, they require an expert dataset of motions. Many such approaches train policies for a specific reference motion or behavioral distribution. While this specialization can be appropriate for some applications, many settings would benefit from a single policy that can, depending on the situation and command, alternate between different types of gaits and skills rather than remain limited to a particular style of motion.

Finally, there are some approaches (Tirinzoni et al., 2025; Li et al., 2025; Luo et al., 2025; Yang et al., 2025) that use a whole dataset of diverse motions as a regularizing factor for a policy. Building on

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

top of them, they create a conditioned policy able to perform multiple different motions depending on the input. Those kinds of efforts are particularly interesting because they can lead to more general, foundational policies able to perform a wide range of tasks.

`Kine2Go` contains 40 diverse motion families, each associated with a dedicated RL tracking policy and represented by 20 stochastic rollouts, for 800 trajectories in total. We hope that our dataset will aid particularly in the last of the aforementioned approaches, where a diverse dataset is the necessary basis. By utilizing multiple data sources and coalescing them into a single pipeline, we were able to produce a comprehensive gait collection, covering many possible movements. The dataset is collected as kinematic motion for a robot, including position, rotation, velocity, angular velocity, joint position, and joint velocities, as well as actions (motor usage) taken by the policy to achieve each trajectory. By conditioning various emerging methods on such a dataset, it could be possible to create a policy capable of natural and diverse locomotion, opening new directions in research on robotics. Such approaches described above were possible in the humanoid space due to the large amount of Motion Capture data. Quadrupedal morphology lacks similar-scale datasets and, as such, has not seen similar efforts. Our work aims to address this gap by introducing an open Go2 state-action dataset spanning diverse motion families.

In addition to the dataset, we also contribute a pipeline that can be easily extended and reused to either gather even more trajectories for the Go2 robot from new data sources or, after minor modifications, transplant our approach to another morphology. Crucially, our pipeline uses the Genesis (Genesis AI Team, 2026) engine, which is fast, easy to work with, and well optimized for consumer-grade GPU hardware. This can further aid researchers willing to experiment on different robots and bring their own data sources.

To demonstrate one of the possible use cases of our dataset, we train a Behavioral Foundation Model (BFM), using `Kine2Go` as a regularizing and evaluation dataset. The resulting model can closely mimic a given trajectory and zero-shot optimize a given reward function, thus demonstrating that `Kine2Go` can support work with quadrupedal behavioral foundation models.

2 Related Work

2.1 Large-scale robot learning datasets

Large-scale datasets have become a central mechanism for improving generalization and reproducibility in robot learning. Early multi-robot resources such as RoboNet aggregated interaction data across several robot platforms and showed that pretraining on shared robot experience can improve adaptation to held-out platforms and environments (Dasari et al., 2019).

More recent manipulation datasets, including BridgeData V2, DROID, and Open X-Embodiment, further demonstrate the value of collecting diverse demonstrations across tasks, objects, environments, and hardware embodiments (Walke et al., 2024; Khazatsky et al., 2025; Collaboration et al., 2025). These datasets are primarily oriented toward visual manipulation and high-level embodied behavior. In contrast, `Kine2Go` focuses on low-level legged locomotion: it provides Go2-compatible kinematic trajectories together with the motor-level actions used to realize them in simulation.

This distinction is important because locomotion policies operate at the motion-control layer. For a quadruped, a policy does not only need to know the task objective or visual scene; it must infer actuator commands from the robot's current dynamic state, including body orientation, velocities, joint configuration, and recent control history. This is why prior motion-imitation datasets and systems separate kinematic reference motions from state-action rollouts: reference motions describe the desired behavior, while rollouts provide the control information needed to reproduce it in simulation or on hardware (Peng et al., 2018). `Kine2Go` therefore complements visual robot datasets by providing Go2-specific kinematic and action trajectories for low-level locomotion learning.

2.2 Kinematic motion datasets

Large-scale motion datasets have become an important substrate for learning motion models and controllers. In human motion learning, datasets such as Human3.6M provide 3D joint trajectories and whole-body motion capture data for pose estimation, motion analysis, and transfer to robot em-

bodiments (Zhu et al., 2023). AMASS further standardized heterogeneous human mocap collections by converting them into a common SMPL body representation, creating a unified motion corpus that has since been used for motion generation, imitation learning, and control (Mahmood et al., 2019). This standardization principle is central to our work: heterogeneous motions become substantially more reusable once they are expressed in a shared embodiment-specific representation.

Comparable resources for animal and quadruped motion are more limited. Animal3D provides 3D pose and shape annotations for diverse mammal species using SMAL parameters, while PFERD provides dense marker-based 3D motion capture for horse locomotion (Xu et al., 2024; Li et al., 2024). These datasets are valuable sources of animal morphology and kinematics, but they are not directly expressed in the morphology, state space, or action space of a robot such as the Unitree Go2. `Kine2Go` addresses this gap by converting heterogeneous quadruped motions into a Go2-compatible representation and augmenting the resulting trajectories with policy-generated motor actions.

2.3 Motion-control and quadruped datasets

Kinematic data describes what motion should be reproduced, but it does not directly provide the motor commands required to realize that motion under robot dynamics. Motion-imitation pipelines, commonly retarget a reference motion to the target morphology, train a controller to track it, and use the resulting simulated rollouts as data. This strategy has been used for agile quadruped locomotion, where animal reference motions were retargeted and imitated by a Laikago robot (Peng et al., 2020).

The closest dataset-level analogue to our setting is MoCapAct, which converts human mocap clips into humanoid datasets containing proprioceptive observations and actions (Wagener et al., 2023).

`Kine2Go` follows the same principle for quadruped locomotion, but targets the Unitree Go2 morphology. Rather than releasing only retargeted reference trajectories, `Kine2Go` provides Go2-compatible rollouts containing root state, joint state, velocities, and policy-generated motor actions. This makes the dataset directly applicable to behavioral cloning, offline control, motion-conditioned policy learning, and evaluation of Go2 locomotion models.

2.4 Behavioral Foundation Models

Behavioral Foundation Models (BFMs) are a recent research direction with only a handful of examples in the literature. Existing models have primarily been developed for humanoid morphologies (Tirinzoni et al., 2025; Li et al., 2025; Luo et al., 2025). The central idea is to learn diverse motion priors that can be conditioned on specific tasks or behaviors. BFMs therefore rely on diverse motion datasets for regularization. Such datasets exist for humanoids but remain limited for quadrupeds. Our work addresses this gap and supports training a BFM for a quadrupedal robot.

3 Dataset gathering pipeline

To easily gather the dataset of motions, we have created a pipeline consisting of 3 main stages:

1. **Kinematic Retargeting**
   To leverage motion capture (MoCap) data from different source morphologies, we implement a retargeting procedure that maps source data onto the specific kinematic structure and degrees of freedom (DoF) of the Unitree Go2 robot. This ensures trajectory alignment while preserving the general motion present in the source.
2. **Reinforcement Learning Motion Imitation**
   The retargeted motions serve as reference trajectories for an RL-based imitation framework. We optimize a distinct control policy for each reference motion, employing a reward function designed to minimize the deviation between the simulated robot's state and the reference trajectory, thus producing robust, physically feasible behaviors.
3. **Trajectory gathering and filtering**
   The optimized policies are deployed in the simulator to generate a set of stochastic rollouts. This process yields data containing both proprioceptive states and corresponding control actions (motor commands). To ensure dataset quality, we filter out unstable trajectories, such as those involving collisions, falls, or significant deviations from the reference motion.

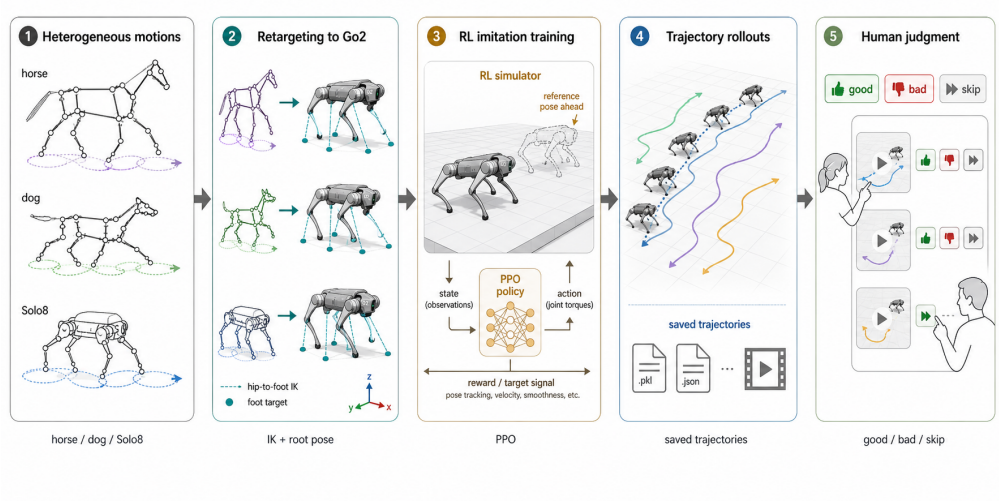


Figure 1: Overview of the entire pipeline

We describe details of each of the above steps in the following subsections and illustrate the entire pipeline in Figure 1. The pipeline follows the general outline of other similar efforts (Peng et al., 2020; Peng, 2025). A key contribution of this work, beyond the dataset itself, is the implementation of this pipeline within Genesis (Genesis AI Team, 2026), a high-performance, GPU-accelerated physics engine, able to run on both consumer-grade and cluster-grade GPUs. Given that our approach necessitates training individual policies for a collection of motions, the massive parallelism and computational efficiency offered by Genesis were critical.

3.1 Kinematic Retargeting

To leverage diverse quadrupedal datasets, we implement a kinematic retargeting pipeline designed to map source motions onto the specific mechanical constraints and degrees of freedom (DoF) of the Unitree Go2 platform. For each data source, we define a separate class that maps the morphology-specific data to full pose information for the Go2 robot. The retargeting process is executed frame-wise. For each timestep in the source trajectory, we extract the relevant target coordinates and resolve the corresponding joint configurations using an inverse kinematics (IK) solver optimized for the Go2's 12-DoF morphology. By treating the retargeted foot positions as task-space objectives, we ensure that the resulting poses preserve the essential gait characteristics of the source data while remaining strictly within the robot's reachable workspace. This modular design facilitates high scalability; incorporating novel data sources requires only the implementation of a specific mapping class. The output of this stage is a time-indexed sequence of Go2-compatible reference poses, which is used as the target for the subsequent imitation-learning stage. Because retargeting is performed frame-wise, this stage does not itself provide control actions or establish dynamic feasibility between frames. The RL imitation stage subsequently realizes the reference sequence under the simulated robot dynamics.

3.2 Motion Imitation

For each kinematic reference trajectory, we optimize a dedicated control policy using a Reinforcement Learning (RL) based imitation framework, following the methodology proposed by (Peng et al., 2020). The task is formulated as a goal-conditioned MDP where the objective is to minimize the tracking error between the simulated agent and the reference motion.

At the onset of each episode, the robot is initialized to the state corresponding to the first frame of the reference trajectory. To ensure heading invariance, a random heading rotation is applied to the global orientation of the reference motion at initialization. The observation space provided to the policy includes the robot's current proprioceptive state (e.g., joint positions, velocities, and base orientation) as well as the reference trajectory state in the future. Specifically, we provide the target poses from the reference trajectory sampled at $t+1, t+2, t+10$, and $t+30$ time steps (see Figure 2).

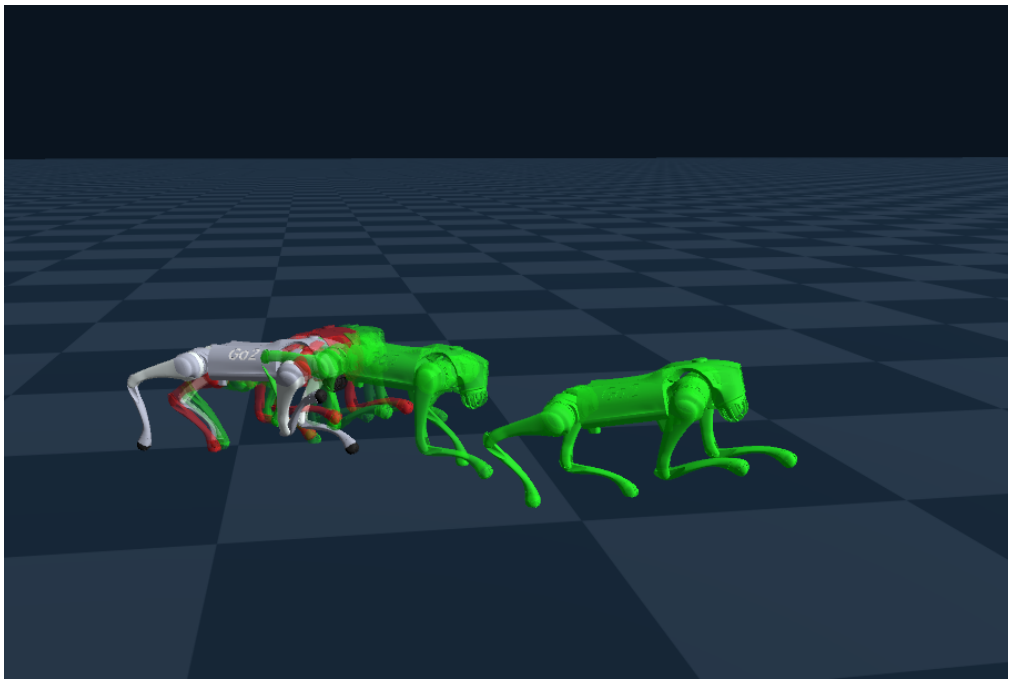


Figure 2: Simulation and observation - The red contour of the robot represents the current state of the reference motion, while four green contours represent the state of the reference motion at 1, 2, 10, and 30 timesteps in the future, which are given to the policy.

The reward function is defined as a weighted sum of several tracking terms, penalizing deviations in root position, linear and angular velocities, and joint-space configurations. We utilize the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017) as implemented in the RSL-RL library (Schwarke et al., 2025). Leveraging the massive parallelism of the Genesis engine, we deploy 8,192 environments simultaneously. This high-throughput configuration enables each policy to be trained for approximately 2 billion environment steps over 10,000 iterations. Since most motions are quite short, we apply periodic wrapping to the reference data, allowing for the generation of trajectories of arbitrary length.

3.3 Trajectory gathering and filtering

Following policy convergence, we execute stochastic rollouts to generate the trajectories of the `Kine2Go` dataset; each rollout is initialized with a randomized heading orientation and a separate seed, introducing variation between policy actions. For a specified number of frames, we record the proprioceptive state and the corresponding control actions of the robot. For each trajectory, we archive a snapshot of the simulator's internal state to facilitate exact reproducibility within the Genesis engine. However, the primary trajectory data is stored in an engine-agnostic format to ensure cross-platform compatibility. The recorded state vector for each frame includes:

1. **Joint Configurations:** Local positions and velocities of the degrees of freedom (DoF) relative to the parent links.
2. **Base Kinematics:** The orientation of the base link represented as a unit quaternion and its associated angular velocity.
3. **Control Actions:** The normalized motor commands (desired offsets w.r.t default positions) issued by the policy.
4. **Global States:** The absolute Cartesian positions and linear velocities of the joint centers in the world frame.

Due to the inherent stochasticity of RL, the deployed policy can occasionally yield sub-optimal behaviors, such as dynamic instability or tracking divergence. To address this, each trajectory is

rendered into a high-resolution video and manually inspected for physical plausibility and adherence to the reference motion. In case of trajectories that are harder to learn, for example, involving complex loops, fast turning, or high speed, it may be necessary to deploy the policy for more episodes to compensate for this difficulty. Our pipeline provides integrated utility scripts to streamline this verification and filtering process. For full details regarding each trajectory, please refer to Appendix B. To further increase the quality of the included data, we have trimmed the resulting trajectories by deleting the first 0.5 seconds, in order for the resulting trajectory not to include the starting "warmup" frames, where the policy tries to get in sync with the reference motion.

Thanks to fast physical simulation and the fact that we can produce a large number of stochastic realizations of each motion family, our pipeline doesn't need vast computational resources and can be successfully used on both small GPU clusters and consumer-grade GPU workstations, thus democratizing the process of robotic data generation for quadrupedal robots.

4 DATA SOURCES FOR KINE2GO

To ensure diverse data, when creating our dataset, we used 4 distinct data sources from motion capture of different morphologies. Since during a single episode a reference motion is cycled multiple times, the first and last frames of the motion must have *similar* joint positions to ensure a smooth transition between cycles. To that end, we had to trim each reference motion, choosing the best start and end frame. This process was done manually by watching the videos of the motions retargeted to the Go2 robot. The complete table outlining all the data sources can be found in Appendix A.

4.1 AI4ANIMATION

Since our methodology is modeled after (Peng et al., 2020), we also used some of the data present in that work. Specifically, we used the motion capture data, gathered from a dog, provided with the code repository. We have used 6 of the trimmed motions used in the original paper, as well as 9 others trimmed manually by us.

4.2 VHDC

This is a MoCap dataset gathered from horses (University of Bonn, 2020). For our research, we restricted our usage to the first dataset provided by this source, namely the Haflinger dataset setup. From this collection, we chose 6 "trot" and 6 "walk" motions originating from the MoCap of two different horses, trimming the trajectories to fit our requirements. Although the full dataset provides more motions, many are quite similar to one another, so we decided to only use a curated subset.

4.3 SOLO8

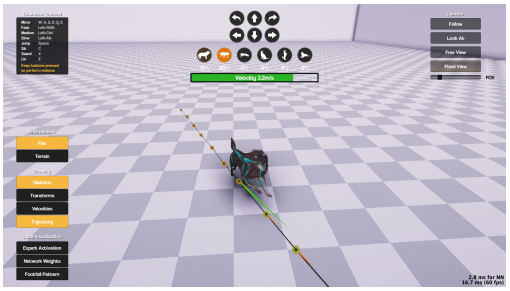
Solo8 is another quadrupedal robot. Its morphology differs from Unitree Go2 in two significant ways: first, it has only 8 degrees of freedom; it has hip and elbow joints, but no shoulder joint, thus the legs move in a plane perpendicular to the torso plane. Secondly, its front and back legs have opposing orientations of elbows, in contrast to Go2, where both front and back elbows are oriented in the same way. An image of the Solo8 robot is presented in Figure 3a. The dataset we used (Li et al., 2023) also provides some motions that are harder to replicate on the Go2 robot, due to morphological differences, for example, walking while crouched close to the ground. As such, data from the Solo8 robot provides not only a source of diverse trajectories, hard to find elsewhere, but also a test for our pipeline.

4.4 AI4ANIMATION INTERACTIVE DEMO

We have also used an interactive simulation from (Zhang et al., 2018). The authors provided a Unity Engine-powered application, where users can steer an animated wolf (see Figure 3b). We modified the original application to capture the kinematic information of the animated dog, and then manually created 7 motions that include walking and running along more complex paths (i.e., ellipses, loops, squares, etc.).



(a) The Solo8 robot has a morphology distinctly different than Unitree Go2. Image sourced from (Chandiramani et al., 2023). Licensed under CC BY 3.0.



(b) The wolf simulator from AI4 Animation. The user can issue commands to the model via keyboard, and thus, we can create complex trajectories.

Figure 3: Solo8 robot and user-controlled wolf simulation.

5 KINE2GO DATASET

Utilizing the pipeline described in Section 3 and the data sources enumerated in Section 4, we created `Kine2Go`, a state-action dataset comprising 40 motion families, each represented by 20 rollouts with randomized initial headings and inherent policy stochasticity. Depending on the type of motion, policies were deployed for between 5 and 20 seconds to capture at least one full cycle of the motion.

`Kine2Go` contains 40 distinct motion families and 800 state-action trajectories in total. We distinguish behavioral diversity, which arises across the 40 motion families, from execution diversity, which arises from 20 stochastic rollouts of the imitation policy associated with each reference motion. The rollouts within a family represent repeated realizations of the same underlying behavior rather than distinct gait classes.

The resulting dataset contains a broad range of locomotive motions, including, but not limited to, walking, running, trotting, turning, and spinning. A detailed breakdown of these 40 motions, alongside a comprehensive description, is provided in Appendix B. We illustrate one trajectory in Figure 4a, while Figure 4b visualizes the example 2D planar coverage achieved across 20 trajectories for a single motion.

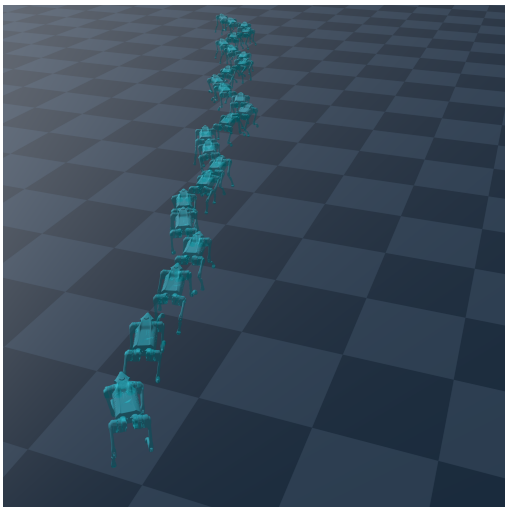
Analogous to the utility of the AMASS or MoCapAct datasets for humanoid research, `Kine2Go` serves as an "off-the-shelf" solution for researchers working with the Unitree Go2 or similar quadrupedal morphologies. By providing an accessible alternative to complex humanoid embodiments, this dataset lowers the barrier to entry for advanced motion research. Consequently, it facilitates rapid experimentation and evaluation of novel methods for agile quadrupedal robotics.

5.1 TRAINING FOUNDATIONAL MODELS ON KINE2GO

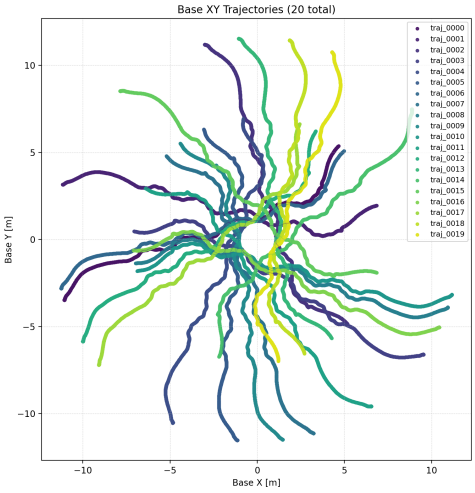
One of the primary motivators for the creation of `Kine2Go` was the prospect of training Behavioral Foundation Models (BFMs). To assess whether our dataset is sufficient for this task, we adapt Meta-Motivo (Tirinzoni et al., 2025) to the Unitree Go2 morphology and train it using `Kine2Go`. In Figure 5 we present the Earth Mover's Distance (EMD) metric (Rubner et al., 2000), measuring the difference between the distribution of the movements in the dataset and those produced by the model. Throughout training, the model learns to closely mimic the target distribution. Performance improves with both the number of motion families (with 20 rollouts per family) and model capacity. A more detailed description of the training protocol and results is available in Appendix E.

6 CONCLUSION

We present `Kine2Go`, a novel and diverse kinematic state-action dataset for the Unitree Go2 robot, comprising 40 unique motion families, each realized by 20 stochastic rollouts, leading to a final 800



(a) Visualization of a single trajectory `ai4_dog_synth_wide_strafe/traj_0000`



(b) X-Y position of the root link of the robot for 20 stochastic trajectories generated from the same policy - `ai4_dog_synth_wide_strafe`

Figure 4: Visualization of example trajectory and trajectories belonging to a single motion family

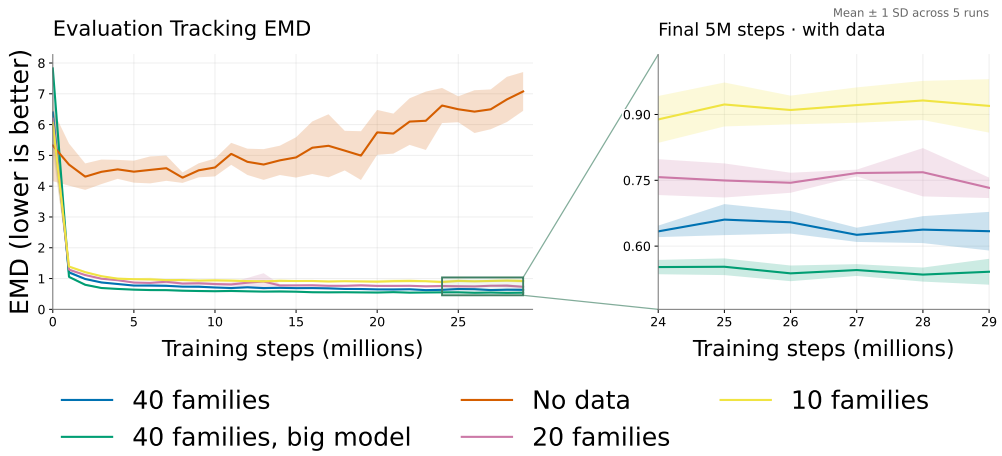


Figure 5: Earth Mover's Distance metric for entire training and final evaluation (lower is better). Each model was trained on 5 seeds; on the curve plot, we present the mean and standard deviation of those runs. We compare the default model with a larger model and ablate over the number of motion families.

trajectories. By aggregating data from varied source morphologies and retargeting them onto a unified quadrupedal embodiment, we provide a broad spectrum of motions in a unified Go2 representation. At the same time, the architecture of the pipeline enables easy extension of the used data sources for specific use cases and future work. While large-scale kinematic datasets have significantly advanced humanoid robotics, the quadrupedal domain has lacked a comparable, open-source resource that bridges the gap between raw motion and actionable RL policies. Training the BFM on `Kine2Go` demonstrates its utility as a behavioral prior in the evaluated setting.

The resulting dataset and accompanying pipeline can support research on foundational locomotion methods for the Unitree Go2 and potentially other quadrupedal platforms. We hope that our contribution will help to discover, evaluate, and deploy novel algorithms leading to significant advancements across Robotics and applied Reinforcement Learning, both in quadrupedal locomotion and more broadly in behavioral foundation models.

7 DISCUSSION

7.1 LIMITATIONS

When creating `Kine2Go`, we strived to cover as much of the state space and feasible motions as possible. However, some behaviors remain underrepresented. In particular, we were unable to reliably imitate several highly aerial or acrobatic jump motions in which all four feet leave the ground. The resulting trajectories were too unstable or inaccurate to include.

Another important category is trajectories that, in Motion Capture, involved a dog sitting. Due to morphological differences between dogs and the Unitree Go2 robot, retargeting a sitting dog results in the robot's elbow joints being below the ground level. To fix this, each sitting trajectory would have to be manually changed, frame by frame, to move each robot's foot to the correct position. It is also worth noting that all the trajectories gathered concern a flat terrain. This is because the dataset is intended for training BFMs, and we believe that it is better to include uneven terrain, as well as other domain randomization methods, during the training of the target BFM model.

In this work, we do not evaluate the trained policies or final BFM model on the real-world robot. Adapting the BFM to physical hardware, potentially using domain randomization and downstream fine-tuning, is an important direction for future work.

Lastly, the Unitree Go2 robot has pressure sensors in each foot. However, we have chosen not to simulate pressure sensors in our simulation, and as such, this data is not present in the dataset.

7.2 FUTURE WORK

The main direction of future work is extending the dataset itself. The main missing parts are described in Section 7.1. The first is expanding coverage of controlled aerial and acrobatic maneuvers, including more challenging jumps and backflips. The second is including gaits and motions in more complex environments, such as ramps and stairs. The BFM results in Figure 5 further suggest that broader motion-family diversity benefits BFM training. We hope that `Kine2Go` will be an ongoing resource that grows through community contributions.

A separate category of future work is the use of the dataset, mainly for training BFMs. We believe that the motions we included can be a solid foundation for general locomotion policies.

Beyond adapting the individual motion policies, an important direction is transferring the BFM itself to physical hardware.

AI USE STATEMENT

In this work, we used generative AI for the following tasks:

- Coding assistance - Coding agents have helped greatly in speeding up the code production for the entire pipeline. The AI-generated code was reviewed by a human and tested to ensure it works. This included code generation, debugging, diagnostics, and tooling.
- Paper writing - The text in the paper was written by the authors, but LLMs were used to make it more polished, readable, and grammatically correct.
- Graphic generation - The pipeline illustration 1 was generated by a coding agent that reviewed the entire pipeline and was asked to produce a graphical illustration of it.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, we publish the code for the entire pipeline (excluding the training of BFM), along with detailed instructions on how it works (including exact data sources for reference motions). The final dataset is also publicly available. The code for BFM training is released in separate repository. The manual review of the resulting trajectories is difficult to reproduce exactly. We describe the review principles in Appendix D, but selection ultimately required human judgment and therefore is not deterministic.

REFERENCES

Vijay Chandiramani, Andrew T. Conn, and Helmut Hauser. Quantifying embodiment towards building better robots based on muscle-driven models. *IOP Conference Series: Materials Science and Engineering*, 1292(1):012004, oct 2023. doi: 10.1088/1757-899X/1292/1/012004. URL https://doi.org/10.1088/1757-899X/1292/1/012004.

Jack Clark and Dario Amodei. Faulty reward functions in the wild, 2016. URL https://openai.com/index/faulty-reward-functions/.

Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, 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 Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, 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, Jay Vakil, 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, Muhammad Zubair Irshad, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, 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 Sonawani, Shubham Tulsiani, 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, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, 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, Zipeng Fu, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and rt-x models, 2025. URL https://arxiv.org/abs/2310.08864.

Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. *CoRR*, abs/1910.11215, 2019. URL http://arxiv.org/abs/1910.11215.

Genesis AI Team. The role of simulation in scalable robotics, genesis world 1.0, and the path forward. *Genesis AI Blog*, May 2026. URL https://www.genesis.ai/blog/the-role-of-simulation-in-scalable-robotics-genesis-world-10-and-the-path-forward.

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, Vitor Guizilini, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Muhammad Zubair Irshad, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, 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, 2025. URL https://arxiv.org/abs/2403.12945.

Chenhao Li, Sebastian Blaes, Pavel Kolev, Marin Vlastelica, Jonas Frey, and Georg Martius. Versatile skill control via self-supervised adversarial imitation of unlabeled mixed motions. In *2023 IEEE international conference on robotics and automation (ICRA)*, pages 2944–2950. IEEE, 2023.

Ci Li, Ylva Mellbin, Johanna Krogager, Senya Polikovsky, Martin Holmberg, Nima Ghorbani, Michael J Black, Hedvig Kjellström, Silvia Zuffi, and Elin Hernlund. The poses for equine research dataset (pferd). *Scientific Data*, 11(1):497, 2024.

Yitang Li, Zhengyi Luo, Tonghe Zhang, Cunxi Dai, Anssi Kanervisto, Andrea Tirinzoni, Haoyang Weng, Kris Kitani, Mateusz Guzek, Ahmed Touati, Alessandro Lazaric, Matteo Pirotta, and Guanya Shi. Bfm-zero: A promptable behavioral foundation model for humanoid control using unsupervised reinforcement learning, 2025. URL https://arxiv.org/abs/2511.04131.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Sirui Chen, Fernando Castañeda, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, Xingye Da, Runyu Ding, Cyrus Hogg, Lina Song, Edy Lim, Eugene Jeong, Tairan He, Haoru Xue, Wenli Xiao, Zi Wang, Simon Yuen, Jan Kautz, Yan Chang, Umar Iqbal, Linxi "Jim" Fan, and Yuke Zhu. Sonic: Supersizing motion tracking for natural humanoid whole-body control, 2025. URL https://arxiv.org/abs/2511.07820.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. Amass: Archive of motion capture as surface shapes, 2019. URL https://arxiv.org/abs/1904.03278.

Xue Bin Peng. Mimickit: A reinforcement learning framework for motion imitation and control. *arXiv preprint arXiv:2510.13794*, 2025. URL https://arxiv.org/abs/2510.13794.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics*, 37(4):1–14, 2018. ISSN 1557-7368. doi: 10.1145/3197517.3201311. URL http://dx.doi.org/10.1145/3197517.3201311.

Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. *arXiv preprint arXiv:2004.00784*, 2020.

Yossi Rubner, Carlo Tomasi, and Leonidas Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:99–121, 11 2000. doi: 10.1023/A:1026543900054.

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

Clemens Schwarke, Mayank Mittal, Nikita Rudin, David Hoeller, and Marco Hutter. Rsl-rl: A learning library for robotics research. *arXiv preprint arXiv:2509.10771*, 2025.

Andrea Tirinzoni, Ahmed Touati, Jesse Farebrother, Mateusz Guzek, Anssi Kanervisto, Yingchen Xu, Alessandro Lazaric, and Matteo Pirotta. Zero-shot whole-body humanoid control via behavioral foundation models, 2025. URL https://arxiv.org/abs/2504.11054.

Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. *CoRR*, abs/2103.07945, 2021. URL https://arxiv.org/abs/2103.07945.

University of Bonn. Vienna horse data collection (vhdc) - horse motion capture data. https://horse.cs.uni-bonn.de/vhdc-home.html, 2020. Accessed: 2026-05-06.

Nolan Wagener, Andrey Kolobov, Felipe Vieira Frujeri, Ricky Loynd, Ching-An Cheng, and Matthew Hausknecht. Mocapact: A multi-task dataset for simulated humanoid control, 2023. URL https://arxiv.org/abs/2208.07363.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale, 2024. URL https://arxiv.org/abs/2308.12952.

Jiacong Xu, Yi Zhang, Jiawei Peng, Wufei Ma, Artur Jesslen, Pengliang Ji, Qixin Hu, Jiehua Zhang, Qihao Liu, Jiahao Wang, Wei Ji, Chen Wang, Xiaoding Yuan, Prakhar Kaushik, Guofeng Zhang, Jie Liu, Yushan Xie, Yawen Cui, Alan Yuille, and Adam Kortylewski. Animal3d: A comprehensive dataset of 3d animal pose and shape, 2024. URL https://arxiv.org/abs/2308.11737.

Ruihan Yang, Zhuoqun Chen, Jianhan Ma, Chongyi Zheng, Yiyu Chen, Quan Nguyen, and Xiaolong Wang. Generalized animal imitator: Agile locomotion with versatile motion prior, 2025. URL https://arxiv.org/abs/2310.01408.

He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. Mode-adaptive neural networks for quadruped motion control. *ACM Trans. Graph.*, 37(4), July 2018. ISSN 0730-0301. doi: 10.1145/3197517.3201366. URL https://doi.org/10.1145/3197517.3201366.

Yue Zhu, Nermin Samet, and David Picard. H3wb: Human3.6m 3d wholebody dataset and benchmark, 2023. URL https://arxiv.org/abs/2211.15692.

A Data sources

Table 1 details the original reference motions used for retargeting across the 40 policies. The file identifiers correspond to the directory structure in the `motion_retargeting/data/` folder of our provided code. Notably, `half_flip_jump` utilizes the same reference motion as `ellipse_walk`. The resulting jumping behavior was an artifact of the policy's inability to accurately track the reference; however, we chose to include this emergent motion in the dataset.

B Motion data details

Table 2 provides a comprehensive list of all motions, including brief descriptions for those that are not self-explanatory. We also specify the rollout lengths for the trajectories sampled for each family.

Policy	Reference File	Start frame	End frame
ai4_dog_canter	dog_run00_joint_pos.txt	430	459
ai4_dog_left_turn	dog_walk09_joint_pos.txt	1085	1124
ai4_dog_pace	dog_walk00_joint_pos.txt	162	201
ai4_dog_right_turn	dog_walk09_joint_pos.txt	2404	2450
ai4_dog_run_00	dog_run00_joint_pos.txt	399	535
ai4_dog_run_02	dog_run02_joint_pos.txt	35	196
ai4_dog_run_04	dog_run04_joint_pos.txt	493	716
ai4_dog_synth_circle_walk	circle_walk.txt	446	806
ai4_dog_synth_eight_walk	eight_walk.txt	0	-1
ai4_dog_synth_ellipse_walk	ellipse_walk.txt	116	1165
ai4_dog_synth_half_flip_jump	ellipse_walk.txt	116	1165
ai4_dog_synth_square_walk	square_walk.txt	29	1087
ai4_dog_synth_tight_strafe	tight_strafe.txt	54	962
ai4_dog_synth_wide_strafe	wide_strafe.txt	121	1080
ai4_dog_trot_00	dog_walk03_joint_pos.txt	448	481
ai4_dog_trot_01	dog_run04_joint_pos.txt	630	663
ai4_dog_walk_00	dog_walk00_joint_pos.txt	101	580
ai4_dog_walk_01	dog_walk01_joint_pos.txt	377	1050
ai4_dog_walk_02	dog_walk02_joint_pos.txt	429	879
ai4_dog_walk_03	dog_walk03_joint_pos.txt	177	545
ai4_dog_walk_04	dog_walk04_joint_pos.txt	234	533
ai4_dog_walk_06	dog_walk06_joint_pos.txt	200	392
solo8_crawl_fast	solo8_motion_data.pt:3	77	119
solo8_crawl_slow	solo8_motion_data.pt:0	25	-1
solo8_jump_forward_a	solo8_motion_data.pt:23	52	114
solo8_jump_forward_b	solo8_motion_data.pt:26	37	112
solo8_scoot_forward	solo8_motion_data.pt:1	69	119
solo8_walk	solo8_motion_data.pt:2	49	84
vhdc_horse1_s1_trot_01	Horse1_M1_trot1_kinematics.csv	13	594
vhdc_horse1_s1_trot_02	Horse1_M1_trot2_kinematics.csv	2	599
vhdc_horse1_s1_trot_03	Horse1_M1_trot3_kinematics.csv	12	567
vhdc_horse1_s1_walk_01	Horse1_M1_walk1_kinematics.csv	9	599
vhdc_horse1_s1_walk_02	Horse1_M1_walk2_kinematics.csv	4	591
vhdc_horse1_s1_walk_03	Horse1_M1_walk3_kinematics.csv	9	598
vhdc_horse1_s2_trot_01	Horse1_M2_trot1_kinematics.csv	7	1144
vhdc_horse1_s2_trot_02	Horse1_M2_trot2_kinematics.csv	3	1173
vhdc_horse1_s2_trot_03	Horse1_M2_trot3_kinematics.csv	10	1194
vhdc_horse1_s2_walk_01	Horse1_M2_walk1_kinematics.csv	16	1145
vhdc_horse1_s2_walk_02	Horse1_M2_walk2_kinematics.csv	27	1153
vhdc_horse1_s2_walk_03	Horse1_M2_walk3_kinematics.csv	40	1137

Table 1: Frame ranges used for reference motion in each policy training; -1 means the last frame.

B.1 Sample selection

The complete dataset, including videos, is approximately 30 GB in size. To facilitate the review process, we have provided a smaller sample dataset. This subset was explicitly chosen to represent a diverse range of behaviors, comprising a total of eight motions (two drawn from each of the four source morphologies). For each motion, we included two trajectories sampled from an RL policy trained on that specific task. The motions selected for this sample are denoted in Table 2.

C Reinforcement Learning details

For the motion imitation component of our pipeline, we reproduced the approach of (Peng et al., 2020) within the Genesis simulation engine. Genesis supports GPU-vectorized, parallel environments, leading to substantially accelerated training times. We utilized the PPO algorithm (Schulman et al., 2017) from the RSL-RL library (Schwarke et al., 2025), with specific hyperparameters detailed in Table 3. Each policy was trained for 10,000 iterations. Because reference motions are typically shorter than a full episode, we implemented reference motion cycling, wherein a motion is repeated

Name	Description	Length	Sample
ai4_dog_canter	Cantering (running) gait	571	✓
ai4_dog_left_turn	Turning left	571	
ai4_dog_pace	Pacing gait	571	
ai4_dog_right_turn	Turning right	571	✓
ai4_dog_run_00		571	
ai4_dog_run_02		571	
ai4_dog_run_04		571	
ai4_dog_synth_circle_walk	Circular walking path	1171	✓
ai4_dog_synth_eight_walk	Figure-eight walking path	1171	
ai4_dog_synth_ellipse_walk	Elliptical walking path	1171	
ai4_dog_synth_half_flip_jump	Horizontal 180°jump	300	✓
ai4_dog_synth_square_walk	Square walking path	1171	
ai4_dog_synth_tight_strafe	Tight strafing motion	1171	
ai4_dog_synth_wide_strafe	Wide strafing motion	960	
ai4_dog_trot_00		571	
ai4_dog_trot_01		571	
ai4_dog_walk_00		571	
ai4_dog_walk_01		571	
ai4_dog_walk_02		571	
ai4_dog_walk_03		571	
ai4_dog_walk_04		571	
ai4_dog_walk_06		571	
solo8_crawl_fast	Crawling close to the ground	571	✓
solo8_crawl_slow	Crawling very close to the ground	571	
solo8_jump_forward_a	Forward jump variant A	571	✓
solo8_jump_forward_b	Forward jump variant B	571	
solo8_scoot_forward	Forward scooting motion	571	
solo8_walk	Walking gait	571	
vhdc_horse1_s1_trot_01		571	✓
vhdc_horse1_s1_trot_02		571	
vhdc_horse1_s1_trot_03		571	
vhdc_horse1_s1_walk_01		571	✓
vhdc_horse1_s1_walk_02		571	
vhdc_horse1_s1_walk_03		571	
vhdc_horse1_s2_trot_01		571	
vhdc_horse1_s2_trot_02		571	
vhdc_horse1_s2_trot_03		571	
vhdc_horse1_s2_walk_01		571	
vhdc_horse1_s2_walk_02		571	
vhdc_horse1_s2_walk_03		571	

Table 2: Motion dataset summary. The length is the number of frames in each of the 20 sampled trajectories.

to fill the episode duration. This requires the start and end frames to be consistent in joint space, although the robot's global position and orientation may vary. Policies were trained on NVIDIA A100 GPUs, with an average training time of 3 hours per motion. Subsequent trajectory sampling and video rendering were conducted on a standard, consumer-grade workstation without GPU acceleration.

C.1 Reward structure

The policy is trained using a weighted imitation reward designed to minimize the discrepancy between the agent's state and a time-aligned reference motion. At each control step t, the environment computes a reference state $\bar{s}_t$ comprising the root pose, root velocity, joint configurations, joint velocities, and end-effector positions. Following the reward structure described in (Peng et al., 2020), the total reward R_t is defined as:

$$R_t = w_q r_q + w_{\dot{q}} r_{\dot{q}} + w_e r_e + w_p r_p + w_v r_v \tag{1}$$

where the reward weights are:

$$w_q = 0.5, \quad w_{\dot{q}} = 0.05, \quad w_e = 0.2, \quad w_p = 0.15, \quad w_v = 0.1. \tag{2}$$

Here q_t and $\bar{q}_t$ denote the robot and reference joint angles, and $\dot{q}_t$ and $\dot{\bar{q}}_t$ denote the corresponding joint velocities.

$$r_q = \exp\left(-5 \left\| q_t - \bar{q}_t \right\|_2^2\right), \tag{3}$$

$$r_{\dot{q}} = \exp\left(-0.1 \left\| \dot{q}_t - \dot{\bar{q}}_t \right\|_2^2\right). \tag{4}$$

The end-effector reward tracks the four feet relative to the robot root. Let $f_{t,k}$ and $\bar{f}_{t,k}$ be the relative foot placement for foot k at timestep t. The end-effector reward is:

$$r_e = \exp\left(-40 \sum_{k=1}^{4} \left\| f_{t,k} - \bar{f}_{t,k} \right\|_2^2\right). \tag{5}$$

The root pose reward consists of separate position and orientation tracking terms. Let p_t and $\bar{p}_t$ be root positions, and let Q_t and $\bar{Q}_t$ be root orientations represented as quaternions. The orientation error is computed from the axis-angle norm of the relative quaternion:

$$Q_{\text{err}} = Q_t \bar{Q}_t^{-1}, \qquad \theta_t = \left\| \text{AxisAngle}(Q_{\text{err}}) \right\|_2. \tag{6}$$

The root pose reward is

$$r_p = \exp\left(-20 \left\| p_t - \bar{p}_t \right\|_2^2\right) + \exp\left(-10 \theta_t^2\right). \tag{7}$$

Finally, the root velocity reward tracks both linear and angular root velocities in the world frame. Let v_t, ω_t be the robot root linear and angular velocities, and $\bar{v}_t, \bar{\omega}_t$ the corresponding reference values:

$$r_v = \exp\left(-2 \left\| v_t - \bar{v}_t \right\|_2^2\right) + \exp\left(-0.2 \left\| \omega_t - \bar{\omega}_t \right\|_2^2\right). \tag{8}$$

D MANUAL REVIEWING PROCESS

The rollouts resulting from trained RL policies were reviewed manually to select those that resemble the motions that the policy was tasked to imitate. We decided to include 20 rollouts gathered from each of the policies. To that end, we gathered 60 rollouts per policy and manually selected 20. The selection process involved watching a clip of the rollout of the robot movement with a visible "ghost" demonstrating the reference movement. This process used human judgement and, as such, is not directly reproducible. The clip was discarded if the robot stumbled, fell, or deviated too much from the movement. Once 20 good clips were found in a given policy, the rest of the clips was discarded. The selection was done by two of the authors, with roughly a 50% split between them. Depending on the complexity of reference motion, the success rate of the policy varied greatly. Some movements, like the `vhdc_horse` family, had a nearly 100% success rate, while in the hardest clip, `ai4_dog_canter`, roughly half of the clips had the robot fall down or stumble.

Category	Parameter	Value
Algorithm	Actor hidden dimensions	[512, 256, 128]
Algorithm	Critic hidden dimensions	[512, 256, 128]
Algorithm	Activation	ELU
Algorithm	Initial action noise std.	1.0
PPO optimization	Learning rate	0.001
PPO optimization	Learning rate schedule	adaptive
PPO optimization	PPO clip parameter	0.2
PPO optimization	Desired KL	0.01
PPO optimization	Entropy coefficient	0.01
PPO optimization	Value loss coefficient	1.0
PPO optimization	Use clipped value loss	true
PPO optimization	Max gradient norm	1.0
PPO optimization	Learning epochs per update	5
PPO optimization	Mini-batches per update	4
PPO optimization	Number of steps	10000
Returns	Discount factor γ	0.99
Returns	GAE parameter λ	0.95
Rollout	Steps per environment	24
Rollout	Maximum iterations	10000
Rollout	Save interval	100
Rollout	Seed	1
Environment	Number of parallel environments	8192
Environment	Episode length	20.0 s
Environment	Control frequency	60 Hz
Environment	Decimation	4
Environment	Action scale	0.25
Environment	Action latency	0.02 s
Environment	Action clipping	100.0
Environment	Policy steps per reference motion step	1
Observation	History length	3
Observation	Observation noise	0.0
Observation	Empirical normalization	none
Imitation reward	Joint rotation scale	0.008333
Imitation reward	Joint velocity scale	0.000833
Imitation reward	End-effector position scale	0.003333
Imitation reward	Root pose scale	0.0025
Imitation reward	Root velocity scale	0.001667
Initialization	Reference-state init probability	0.9
Initialization	Perturb initial state	true
Initialization	Joint position randomization range	0.3
Initialization	Base position randomization range	1.0
Initialization	Randomize base yaw	true
Initialization	Randomize trajectory heading	true
Termination	Distance threshold	1.0 m
Termination	Rotation threshold	1.5708 rad

Table 3: PPO training hyperparameters used for motion imitation.

E BFM MODEL

E.1 ARCHITECTURE AND TRAINING

We closely follow (Tirinzoni et al., 2025), swapping the environment from HumEnv to our own Genesis-based implementation centered around Unitree Go2. We reuse the implementation of the entire Meta Motivo pipeline, with small adjustments. The algorithm enabling the training is FB-CPR (Forward-Backward representations with Conditional Policy Regularization)(Tirinzoni et al.,

2025). The Forward-Backward algorithm of Touati and Ollivier (2021) disentangles the dynamics of the environment and rewards, thus creating a model that can achieve a given state, follow a given trajectory, or zero-shot optimize a given reward function. FB-CPR builds on top of it by regularizing the policy behavior with a reference motion dataset.

We use a latent-conditioned neural architecture with a 256-dimensional latent variable z. The policy combines embeddings of the current 253-dimensional proprioceptive observation and z to produce 12 joint actions; accompanying forward and backward networks map between state–action/latent representations, with critic and discriminator networks used during training. The baseline uses standard MLPs with width 1,024 and two hidden layers, totaling 23.0M parameters across all non-target network components (12.5M in the forward, backward, and policy networks). In the large model, these networks are replaced by residual MLPs of width 2,048 with 12 residual blocks, yielding 284.8M parameters in total (169.8M in the forward, backward, and policy networks). Training additionally maintains target copies of the forward, backward, and critic networks, bringing the checkpoint sizes to 39.5M and 510.8M parameters for the baseline and large models, respectively.

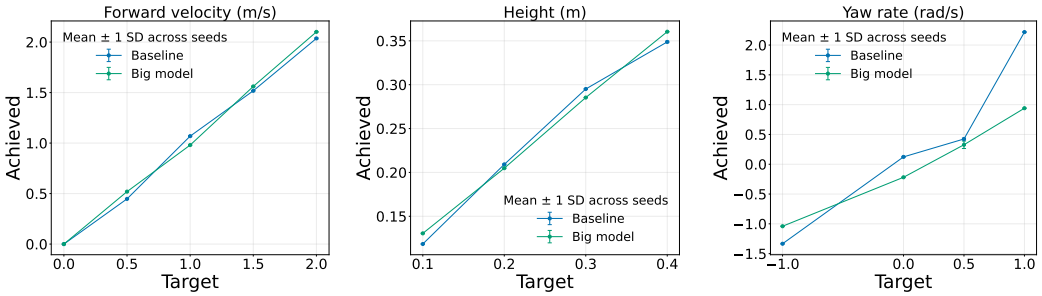
E.2 Results

Since the field of BFMs is not yet well developed, evaluation is not formalized and standardized. Following (Tirinzoni et al., 2025), we evaluate our model in two categories: motion tracking (can the model follow a given trajectory) and reward tracking (can the model zero-shot optimize a given reward function).

One useful metric is the Earth Mover's distance, measuring how far the distribution of movements produced by the model is from movements taken from the dataset. This metric is presented in Figure 5. A motion dataset is necessary for this regularization scheme. We ablate over the number of motion families: the 10-family and 20-family datasets are subsampled from the full 40-family dataset, with 20 rollouts retained for each included family. Reducing the number of motion families substantially degrades performance. We additionally train a larger model, which achieves better performance. These findings are consistent with analogous experiments in the original Motivo paper (Tirinzoni et al., 2025).

We report the motion tracking modality in tables 4 and 5 for the baseline and large models, respectively. In addition to EMD, we report the success fraction and whole-body link velocity error. The success fraction is the proportion of frames whose Euclidean distance from the reference pose is at most 1.25. This distance is computed over a concatenated heading-aligned representation containing base height, the base-relative positions of 16 non-base links, and the six-dimensional orientations of all 17 links. It is therefore a **combined position-and-orientation error, rather than a distance in metres or an average error per link**. The whole-body link velocity error measures differences in frame-to-frame changes of the base-relative, heading-aligned link positions. Displacements are divided by the timestep to obtain velocities, and the Euclidean norm of the velocity discrepancy across all 48 position coordinates is averaged over time and reported in mm/s. Thus, this value **aggregates velocity error across the body; it is neither a per-link average nor a sum over the trajectory**. We report the mean and sample standard deviation of these trajectory-level metrics across the 20 trajectories in each motion family.

One of the interesting properties of the FB-CPR algorithm is the disentanglement of the environment dynamics from rewards. Thus, the algorithm has the capacity to zero-shot optimize a given reward function it has never seen before. It does so by annotating a large corpus of transitions with the given reward function, and interpolating in the latent space of those transitions weighted by annotated rewards. In our evaluation, we used the entire dataset as the corpus. We have tested how well the model is able to follow a given forward velocity, target body height, and yaw turning speed.



(a) Forward velocity (b) Base link height (c) Yaw change rate

Figure 6: Zero-shot reward-conditioned control. We evaluate one baseline and one large model on target forward velocity, base height, and yaw rate. Each target is evaluated with 10 seeds; the plots show the mean and standard deviation.

Table 4: Tracking results for the baseline model. Values are mean ± standard deviation.

Family	Success Fraction	EMD	Whole body link velocity error (mm/s)
ai4_dog_canter	0.213 ± 0.19	2.72 ± 3.1	5897 ± 1007
ai4_dog_left_turn	1 ± 0	0.378 ± 0.0033	1007 ± 21
ai4_dog_pace	0.914 ± 0.11	0.351 ± 0.03	1601 ± 387
ai4_dog_right_turn	0.99 ± 0.011	0.466 ± 0.0088	1729 ± 22
ai4_dog_run_00	0.201 ± 0.085	0.699 ± 0.06	6995 ± 796
ai4_dog_run_02	0.304 ± 0.26	1.51 ± 2.1	6104 ± 2045
ai4_dog_run_04	0.387 ± 0.12	0.454 ± 0.042	3700 ± 582
ai4_dog_synth_circle_walk	0.987 ± 0.025	0.249 ± 0.0042	914 ± 109
ai4_dog_synth_eight_walk	0.823 ± 0.016	0.413 ± 0.004	1976 ± 37
ai4_dog_synth_ellipse_walk	0.857 ± 0.033	0.255 ± 0.01	1841 ± 161
ai4_dog_synth_half_flip_jump	0.0804 ± 0.25	6.91 ± 2.2	3758 ± 671
ai4_dog_synth_square_walk	0.912 ± 0.035	0.297 ± 0.0051	1726 ± 116
ai4_dog_synth_tight_strafe	0.924 ± 0.011	0.429 ± 0.007	1590 ± 44
ai4_dog_synth_wide_strafe	0.983 ± 0.01	0.327 ± 0.0052	1251 ± 47
ai4_dog_trot_00	0.432 ± 0.24	0.733 ± 1.6	3596 ± 810
ai4_dog_trot_01	0.745 ± 0.21	0.711 ± 1.4	2142 ± 637
ai4_dog_walk_00	0.963 ± 0.076	0.3 ± 0.018	1264 ± 298
ai4_dog_walk_01	0.98 ± 0.011	0.332 ± 0.0069	1146 ± 57
ai4_dog_walk_02	0.935 ± 0.02	0.331 ± 0.01	1308 ± 67
ai4_dog_walk_03	0.846 ± 0.11	0.356 ± 0.027	1913 ± 323
ai4_dog_walk_04	0.746 ± 0.18	0.33 ± 0.023	2383 ± 636
ai4_dog_walk_06	0.556 ± 0.26	0.418 ± 0.061	3107 ± 955
solo8_crawl_fast	0.991 ± 0.015	0.357 ± 0.013	1939 ± 399
solo8_crawl_slow	0.993 ± 0.0098	0.527 ± 0.023	1316 ± 31
solo8_jump_forward_a	0.43 ± 0.29	0.404 ± 0.02	4566 ± 1122
solo8_jump_forward_b	0.983 ± 0.043	0.393 ± 0.013	1414 ± 273
solo8_scoot_forward	0.889 ± 0.12	0.409 ± 0.013	2586 ± 520
solo8_walk	0.99 ± 0.023	0.413 ± 0.0088	1180 ± 80
standing	1 ± 0	0.108 ± 0.0029	19 ± 0.3
vhdc_horse1_s1_trot_01	0.984 ± 0.025	0.388 ± 0.019	1410 ± 89
vhdc_horse1_s1_trot_02	0.999 ± 0.0047	0.385 ± 0.011	1178 ± 58
vhdc_horse1_s1_trot_03	0.977 ± 0.037	0.343 ± 0.0099	1182 ± 162
vhdc_horse1_s1_walk_01	1 ± 0	0.362 ± 0.012	1000 ± 22
vhdc_horse1_s1_walk_02	1 ± 0	0.35 ± 0.0064	998 ± 33
vhdc_horse1_s1_walk_03	0.999 ± 0.0021	0.37 ± 0.0068	933 ± 43
vhdc_horse1_s2_trot_01	1 ± 0.0016	0.312 ± 0.012	701 ± 41
vhdc_horse1_s2_trot_02	1 ± 0	0.287 ± 0.021	1129 ± 30
vhdc_horse1_s2_trot_03	1 ± 0	0.322 ± 0.017	699 ± 27
vhdc_horse1_s2_walk_01	1 ± 0.00039	0.331 ± 0.016	640 ± 33
vhdc_horse1_s2_walk_02	1 ± 0	0.284 ± 0.014	592 ± 28
vhdc_horse1_s2_walk_03	1 ± 0	0.305 ± 0.008	606 ± 30

Table 5: Tracking results for the large model. Values are mean ± standard deviation.

Family	Success Fraction	EMD	Whole body link velocity error (mm/s)
ai4_dog_canter	0.453 ± 0.28	1.69 ± 2.5	4411 ± 1189
ai4_dog_left_turn	1 ± 0	0.312 ± 0.0064	880 ± 24
ai4_dog_pace	0.834 ± 0.22	0.247 ± 0.017	1746 ± 752
ai4_dog_right_turn	1 ± 0	0.367 ± 0.0042	1200 ± 32
ai4_dog_run_00	0.427 ± 0.19	0.551 ± 0.12	4756 ± 1045
ai4_dog_run_02	0.265 ± 0.19	1.35 ± 2.1	6234 ± 1647
ai4_dog_run_04	0.813 ± 0.12	0.385 ± 0.039	2235 ± 442
ai4_dog_synth_circle_walk	0.713 ± 0.084	0.217 ± 0.007	2281 ± 291
ai4_dog_synth_eight_walk	0.971 ± 0.0081	0.3 ± 0.0056	1318 ± 64
ai4_dog_synth_ellipse_walk	0.948 ± 0.011	0.203 ± 0.0073	1375 ± 95
ai4_dog_synth_half_flip_jump	0.0873 ± 0.27	6.94 ± 2.2	3105 ± 435
ai4_dog_synth_square_walk	0.966 ± 0.024	0.243 ± 0.0053	1470 ± 118
ai4_dog_synth_tight_strafe	0.954 ± 0.0087	0.327 ± 0.0055	1325 ± 44
ai4_dog_synth_wide_strafe	0.997 ± 0.0033	0.246 ± 0.0027	1101 ± 52
ai4_dog_trot_00	0.701 ± 0.15	0.301 ± 0.013	2780 ± 554
ai4_dog_trot_01	0.803 ± 0.12	0.261 ± 0.051	1987 ± 552
ai4_dog_walk_00	0.991 ± 0.013	0.217 ± 0.0094	1122 ± 147
ai4_dog_walk_01	0.999 ± 0.0038	0.246 ± 0.004	879 ± 45
ai4_dog_walk_02	0.959 ± 0.027	0.26 ± 0.012	1414 ± 91
ai4_dog_walk_03	0.91 ± 0.081	0.315 ± 0.028	1614 ± 318
ai4_dog_walk_04	0.944 ± 0.081	0.307 ± 0.026	1544 ± 359
ai4_dog_walk_06	0.901 ± 0.13	0.354 ± 0.057	1734 ± 548
solo8_crawl_fast	0.964 ± 0.063	0.348 ± 0.014	2235 ± 325
solo8_crawl_slow	0.995 ± 0.015	0.344 ± 0.017	1147 ± 54
solo8_jump_forward_a	0.695 ± 0.29	0.302 ± 0.026	3285 ± 1157
solo8_jump_forward_b	0.942 ± 0.19	0.273 ± 0.013	1343 ± 815
solo8_scoot_forward	0.367 ± 0.11	0.315 ± 0.015	4432 ± 486
solo8_walk	0.763 ± 0.049	0.317 ± 0.012	1931 ± 90
standing	1 ± 0	0.0854 ± 0.0023	15 ± 0.2
vhdc_horse1_s1_trot_01	0.987 ± 0.02	0.227 ± 0.012	1197 ± 160
vhdc_horse1_s1_trot_02	0.974 ± 0.041	0.211 ± 0.0096	1122 ± 123
vhdc_horse1_s1_trot_03	0.984 ± 0.023	0.211 ± 0.0098	1016 ± 136
vhdc_horse1_s1_walk_01	0.998 ± 0.0031	0.306 ± 0.015	1251 ± 29
vhdc_horse1_s1_walk_02	1 ± 0.00039	0.279 ± 0.014	1233 ± 46
vhdc_horse1_s1_walk_03	0.994 ± 0.0063	0.305 ± 0.0092	1195 ± 49
vhdc_horse1_s2_trot_01	0.999 ± 0.0027	0.238 ± 0.0096	659 ± 45
vhdc_horse1_s2_trot_02	1 ± 0	0.313 ± 0.016	1081 ± 47
vhdc_horse1_s2_trot_03	1 ± 0.00078	0.252 ± 0.0073	645 ± 43
vhdc_horse1_s2_walk_01	1 ± 0	0.269 ± 0.01	596 ± 36
vhdc_horse1_s2_walk_02	1 ± 0	0.265 ± 0.0063	577 ± 23
vhdc_horse1_s2_walk_03	1 ± 0	0.257 ± 0.0054	599 ± 31