Learning Robot Locomotion from Diverse Datasets

Lu Liu*[1], Michael Drolet*[1], Oleg Arenz[1], Jan Peters[1]
[1]Department of Computer Science, TU Darmstadt, Germany
{*lu.liu1, michael.drolet, oleg.arenz, jan.peters*}@*tu-darmstadt.de* *equal contribution

1 Introduction

Quadruped robots have gained much attention in recent years [3, 7, 15]. Meanwhile, the Generative Pre-trained Transformer (GPT) models have achieved remarkable success in natural language processing. The abundance of online datasets offers a promising avenue to extend this framework to robotics by representing motion sequences as tokenized data. In this work, we employ a GPT-style network to generate motion sequences of arbitrary length conditioned on a given gait and duration. We investigate the ability to generate natural and bio-inspired locomotion using data-driven techniques and recent learning-based architectures. When integrated with a low-level policy, this approach enables the robot to demonstrate diverse and natural gaits in simulation while preserving the gait style encoded in the various collected datasets.

The first challenge we address is motion retargeting, which involves transferring motion sequences from sources with different sizes and morphologies (e.g., a horse or another robot like the Solo8 [6]) to the target quadrupedal robot, a Unitree Go2 or Unitree A1. This task is challenging due to significant differences in kinematics, dynamics, and actuation constraints between the source and target systems. The second challenge lies in effectively representing motion sequences while preserving their underlying structure. To address this, we adopt the framework from [10], which employs a Vector Quantized Variational Autoencoder (VQ-VAE) [11] to map motion sequences into discrete latent codes. Compared to traditional Variational Autoencoders (VAEs) [8], which uses a continuous bottleneck, VQ-VAE uses a discrete latent space. This learned space is more bit-efficient compared to the traditional VAE and can be used for downstream tasks such as next-token prediction. For the trajectory generation process, we adopt a transformer [16], which enables us to predict the token sequence autoregressively. In order to synthesize this generated trajectory on the robot, we adopt a reinforcement learning approach similar to DeepMimic [12], which employs a time-step-based tracking reward to imitate reference trajectories.

In contrast to similar methods that rely on a single dog dataset [1, 5, 7], our training dataset contains a diverse and extensive quadruped dataset, such as MATLAB trot data [4], horse motion capture data [2], and solo8 data [9]. Finally, we showcase the gait generated by the high-level policy (in combination with the low-level policy) on the Unitree Go2 and the Unitree A1 robot in simulation.

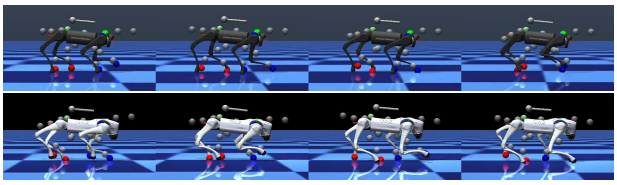

Figure 1: Retargeted Motion: Dog pace gait on Unitree A1 (top) and Unitree Go2 (bottom). The original modalities are given by the colored spheres: front feet (blue), rear feet (red), and withers and hip (green).

2 Methodology

Motion Retargeting. The retargeting method we use is similar to [1]. We start with a keypoint position of the target system and source system, denoted as $\mathbf{p}^{\text{tar}}$, $\mathbf{p}^{\text{src}}$ respectively. In a given kinematic tree, the directional vector between the key point j and its parent can be described as: $\mathbf{d}_j^{\text{src}} = \mathbf{p}_j^{\text{src}} - \mathbf{p}_{\mathscr{P}(j)}^{\text{src}}$, where $\mathscr{P}(\cdot)$ represents the parent function in the kinematic tree. The keypoint in the target system is then defined as: $\mathbf{p}_j^{\text{tar}} = \mathbf{p}_{\mathscr{P}(j)}^{\text{tar}} + \alpha \mathbf{d}_j^{\text{src}}$ where α is a scaling factor determined by the stance height ratio between the target system and the source system. We extract keypoints (e.g., withers, hip, toes) from horse and dog skeletons, and these skeletons are then scaled to the skeleton of the robot. The robot base position is the midpoint between the withers and the hip, and its orientation is derived from the hip to the withers vector. The foot positions are determined by directional vectors from the withers or hip to the toes, and inverse kinematics is used to obtain the joint configurations.

Motion Synthesis. The latent variables given by VQ-VAE are represented by a set of discrete codes $\mathbf{e}$, which collectively form our codebook. Each index i in the codebook points to a unique code $\mathbf{e}_i$. This discrete representation is achieved through vector quantization, a process that maps the continuous latent variables $z_e(\mathbf{x})$ from the encoder to its closest discrete code $\mathbf{e}_k$ in the codebook. The decoder then reconstructs the input $\mathbf{x}$ from the quantized latent representation $z_q(\mathbf{x}) = \mathbf{e}_k$, where $k = \arg\min_i \|z_e(\mathbf{x}) - \mathbf{e}_i\|_2$. Note that the vector quantization process is non-differentiable, so we may only use the argmin operation during the forward pass but may bypass it during the backward pass using a stop gradient operation. Once the VQ-VAE is trained, generating motion sequences reduces to generating a sequence of action vocabulary indices. The GPT network autoregressively predicts the indices using self-attention and causal masking, conditioned on a given gait label and duration [10].

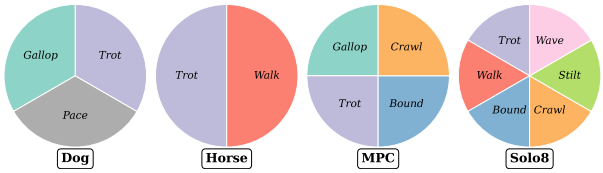

Figure 2: Overview of the dataset distribution and diversity.

Motion Tracking. In order to execute the generated trajectories, we train a low-level reinforcement learning policy using DeepMimic [12]. Let τ be a $T \times D$ dimensional reference gait trajectory produced from the high-level policy, containing the joint positions and velocities, position and orientation of the trunk base, and linear and angular velocities of the trunk base. The tracking reward can be summarized as:

$$r_t = \sum_{d=1}^{D} \exp(-w_d(\tau_t^{d*} \ominus \tau_t^d)^2). \quad (1)$$

That is, the sum of exponentiated negative square differences between the reference feature d at time t, τ_t^{d*}, and the currently observed feature, τ_t^d. The total return using this reward is bounded by 100, as can be seen in Figure 3. The weights for each feature are given by w_d. We use $\ominus$ to denote the difference of features since the rotation features lie in $\mathbf{so}(3)$. The exponential form ensures that the reward decreases smoothly as the robot's differs from the reference state. This policy is trained using Proximal Policy Optimization (PPO) [13].

3 Experiments

We collect several datasets from publicly available sources to evaluate our motion retargeting algorithm, including motion capture data from dogs [17], horses [2], and various robot platforms [4, 9]. Each combination of motion source and gait contains 10 trajectories, each approximately 2 seconds long, and all trajectories are resampled at a frequency of 50 Hz. An overview is shown in Figure 2. At the beginning of each episode during training, the environment randomly selects a reference trajectory from the reference buffer provided by the high-level policy. Additionally, two one-hot encodings are used in the observation space to indicate the gait type and expert platform. To evaluate the quality and diversity of the generated samples, we adopt metrics suggested by [14]. First, we train a transformer-based classifier C on a training dataset $\mathscr{D}_t$, denoted as $C^{\mathscr{D}_t}$. This classifier is trained to minimize the loss between the predicted gait label (e.g., dog-trot or horse-walk) and its true label for all trajectories in $\mathscr{D}_t$. Next, we evaluate the accuracy of this classifier on a different dataset $\mathscr{D}_v$ by taking the expectation of the classification accuracy with respect to the dataset: i.e., $\text{acc}(\mathbb{E}_{\mathbf{x}\sim\mathscr{D}_v}[C^{\mathscr{D}_t}(\mathbf{x})])$. Therefore, the metric, $\text{acc}(\mathbb{E}_{\mathbf{x}\sim\pi_E^{\text{val}}}[C^{\pi}(\mathbf{x})])$, reflects the diversity of the generated samples from the policy (where the superscript "val" denotes the validation set and the subscript "E" denotes expert). The dataset from the policy includes 24 generated trajectories per gait, each approximately 2 seconds long, totaling around 360 trajectories. The evaluation metrics in Table 1 highlight the performance of the high level policy, among other interesting interpretations described in [14]. The results suggest that our high level policy can produce trajectories that closely resemble the expert while maintaining sufficient diversity.

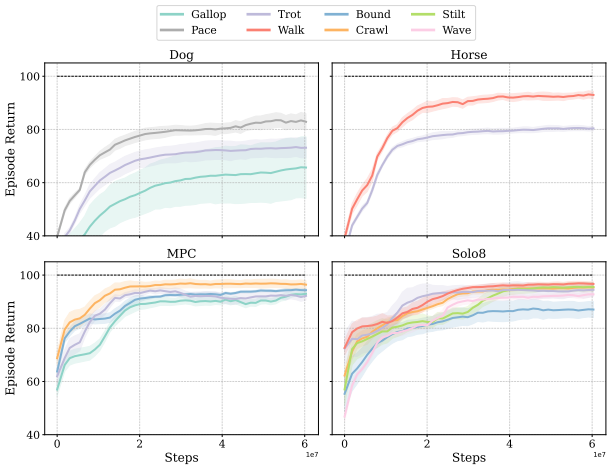

Figure 3: Low Level Policy Tracking Performance.

Dataset	$\text{acc}(\mathbb{E}_{\mathbf{x}\sim\pi_E^{\text{val}}}[C^{\pi_E}(\mathbf{x})])$	$\text{acc}(\mathbb{E}_{\mathbf{x}\sim\pi}[C^{\pi_E}(\mathbf{x})])$	$\text{acc}(\mathbb{E}_{\mathbf{x}\sim\pi_E^{\text{val}}}[C^{\pi}(\mathbf{x})])$
All	0.933	0.736	0.833
Dog	0.889	0.611	1.000
Horse	1.000	0.708	0.778
MPC	0.867	0.833	0.933
Solo8	0.963	0.743	0.741

Table 1: Evaluation Metrics for Sample Quality and Diversity by Different Motion Source

4 Discussion

While results presented so far can be utilized for deploying locomotion policies in the simulator, there are still steps needed to transfer this to the real system. We perform an ablation study to train the low level policy with domain randomization (to facilitate sim2real transfer) and can achieve visually-similar gaits in many cases; however, future work includes transferring this policy to the real system. Other promising directions include using human-based datasets and using the learned policies to control exoskeletons or humanoid robots. One limitation of our work includes the inability to command the robot, as it is trained to mimic the dataset. To achieve this, we can introduce goal contexts during training and increase the diversity of the dataset to include demonstrations of locomotion in multiple directions. In summary, we incorporate motion retargeting, motion synthesis, and motion tracking into a unified framework and demonstrate that it can effectively map diverse locomotion behaviors to different robot platforms. These results also demonstrate the feasibility of using a modular approach and the potential for scaling the method to larger datasets. Furthermore, we showcase the potential to learn a compact latent representation that transfers across different embodiments, each of which have unique kinematics.

5 Acknowledgements

This research was supported partly by the German Research Foundation (DFG) within RTG 2761 LokoAssist under grant no. 450821862.

References

[1] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. "Learning Agile Robotic Locomotion Skills by Imitating Animals". In: Robotics: Science and Systems Foundation, July 12, 2020.

[2] University of Bonn. *HORSE Project*. 2024.

[3] Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. *Extreme Parkour with Legged Robots*. 2023.

[4] Yanran Ding, Abhishek Pandala, Chuanzheng Li, Young-Ha Shin, and Hae-Won Park. "Representation-Free Model Predictive Control for Dynamic Motions in Quadrupeds". In: *IEEE Transactions on Robotics* 37.4 (Aug. 2021), pp. 1154–1171.

[5] Alejandro Escontrela, Xue Bin Peng, Wenhao Yu, Tingnan Zhang, Atil Iscen, Ken Goldberg, and Pieter Abbeel. *Adversarial Motion Priors Make Good Substitutes for Complex Reward Functions*. Mar. 28, 2022.

[6] Felix Grimminger, Avadesh Meduri, Majid Khadiv, Julian Viereck, Manuel Wuthrich, Maximilien Naveau, Vincent Berenz, Steve Heim, Felix Widmaier, Thomas Flayols, Jonathan Fiene, Alexander Badri-Sprowitz, and Ludovic Righetti. "An Open Torque-Controlled Modular Robot Architecture for Legged Locomotion Research". In: *IEEE Robotics and Automation Letters* 5.2 (Apr. 2020), pp. 3650–3657.

[7] Lei Han, Qingxu Zhu, Jiapeng Sheng, Chong Zhang, Tingguang Li, Yizheng Zhang, He Zhang, Yuzhen Liu, Cheng Zhou, Rui Zhao, Jie Li, Yufeng Zhang, Rui Wang, Wanchao Chi, Xiong Li, Yonghui Zhu, Lingzhu Xiang, Xiao Teng, and Zhengyou Zhang. "Lifelike agility and play in quadrupedal robots using reinforcement learning and generative pre-trained models". In: *Nature Machine Intelligence* 6.7 (July 2024), pp. 787–798.

[8] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2022.

[9] 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*. Feb. 11, 2023.

[10] Thomas Lucas, Fabien Baradel, Philippe Weinzaepfel, and Grégory Rogez. *PoseGPT: Quantization-based 3D Human Motion Generation and Forecasting*. Oct. 19, 2022.

[11] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu koray. "Neural Discrete Representation Learning". In: vol. 30. Curran Associates, Inc., 2017.

[12] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van De Panne. "DeepMimic: example-guided deep reinforcement learning of physics-based character skills". In: *ACM Transactions on Graphics* 37.4 (Aug. 31, 2018), pp. 1–14.

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

[14] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. *How good is my GAN?* July 25, 2018.

[15] Laura Smith, J. Chase Kew, Tianyu Li, Linda Luu, Xue Bin Peng, Sehoon Ha, Jie Tan, and Sergey Levine. *Learning and Adapting Agile Locomotion Skills by Transferring Experience*. Apr. 19, 2023.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. *Attention Is All You Need*. Aug. 1, 2023.

[17] He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. "Mode-adaptive neural networks for quadruped motion control". In: *ACM Transactions on Graphics* 37.4 (Aug. 31, 2018), pp. 1–11.