# Dynamics-Driven Policy Transfer to Heavy Wheel-Legged Robots via Imitation and Optimization

Chengleng Han, Lin Xu *, and Changshun Huang

Hubei Key Laboratory of Advanced Technology for Automotive Components,
Wuhan University of Technology, Wuhan, China
Email: hanchengleng@whut.edu.cn (C.H.); xulin508@whut.edu.cn (L.X.); 347757@whut.edu.cn (C.Hu.)
*Corresponding author

***Abstract*—Policy transfer is an efficient approach for developing specific robots. Its effectiveness depends on high-quality imitation datasets and a stable learning process. However, substantial differences in geometry and dynamics between source and target robots pose challenges. Purely kinematics-driven mapping methods and manual parameter tuning often fail to maintain kinematic-dynamic consistency. In this study, we transfer control policies from the quadruped robot Unitree Go1 to our self-developed heavy wheel-legged robot Tiangou. We propose a Consistency-Aware Retargeting (CAR) method. This extends conventional inverse kinematics by adding dynamic consistency constraints. Using motion data from Go1's Model Predictive Controller (MPC), CAR generates a reference dataset for Tiangou. We then integrate Bayesian Optimization (BO) into the imitation learning framework. This enables autonomous tuning of policy model structures and optimization hyperparameters. Experiments show that CAR reduces foot-end position errors, mitigates joint angular velocity fluctuations, and decreases foot-end slippage. Moreover, Bayesian optimization improves sample efficiency and training stability. These contributions establish a practical foundation for policy transfer across heterogeneous robotic platforms.**

***Keywords*—imitation learning, motion retargeting, Bayesian optimization, policy transfer, wheel-legged robot**

## I. Introduction

Legged robots offer key advantages through discrete footholds [1, 2]. They excel in adapting to complex terrain and provide superior mobility. This makes them ideal for applications like disaster response [3], field exploration [4], and agricultural operations [5]. In recent years, several quadruped platforms have emerged, which have accelerated advancements in legged robot control [6]. However, pure leg structures limit mobility and energy efficiency due to the actuator's reciprocating oscillation mode [7]. To overcome this, researchers have developed hybrid wheel-legged designs, such as mounting driving wheels at the foot end [8, 9]. Thus, they combine the high-speed mobility of wheeled robots with the terrain adaptability of legged robots. Yet, such designs often restrict load capacity due to joint mechanical limits and require joint actuators for propulsion in wheeled-driven mode, which reduces efficiency and scalability. The wheeled robots dominate practical applications over legged ones. Therefore, we argue that wheel-legged robots should prioritize wheels for mobility and load-bearing. Legs should mainly provide terrain adaptability. Based on this principle, we designed Tiangou, a novel wheel-legged robot, as shown in Fig. 1. It attaches four mechanical legs to an Unmanned Ground Vehicle (UGV) chassis. Tiangou features 14 actuators: 12 for the legs and 2 for the wheels, utilizing high-torque servo motors (e.g., with a peak torque of 57.5 Nm) and integrated encoders for precise feedback. Two caster wheels connect to the frame via a single trailing suspension system, which absorbs vibrations in wheel mode. Each leg uses a 3 Degrees of Freedom (3-DoF) serial mechanism for walking, supported by an Inertial Measurement Unit (IMU) for pose estimation. The wheel and legged modes operate via independent control loops. In wheel mode, Tiangou supports a 150 kg payload and reaches a speed of 50 km/h. In legged mode, the wheels do not drive; instead, they extend the body length by over 0.4 m to avoid interference. This results in a body-leg ratio of 1.83 for Tiangou versus 0.97 for Go1. Besides, Tiangou weighs 80 kg, a 7.7× mass disparity compared to Go1's 10.4 kg. These complicate legged locomotion control design and optimization.

To address these control issues on heavy platforms like Tiangou, we utilize Imitation Learning (IL). This method has proven effective for robot control, particularly in facilitating the efficient transfer of policies across different platforms. For example, GenLoco [10] uses a size factor $\alpha$ (sampled from [0.8, 1.2]) for transfer among similar-sized quadrupeds. However, it ignores cases like Tiangou, with a large body and short legs. Such mismatches amplify dynamic differences, making GenLoco unsuitable—it fails to prevent joint overload or instability during transfer from lightweight to heavy robots. Moreover, training results

Manuscript received September 28, 2025; revised October 14, 2025; accepted November 24, 2025; published January 9, 2026.

depend heavily on model parameters and algorithm hyperparameters. Standard tuning practices for general quadrupeds do not apply to specialized platforms, such as Tiangou. Manual tuning lacks sample efficiency and training stability.

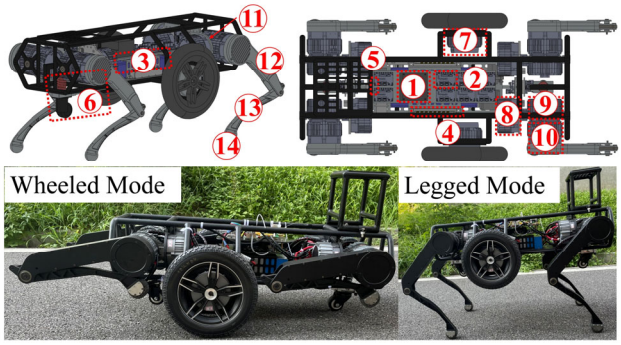


Fig. 1. Wheel-legged robot Tiangou with labeled components. (1) main control unit; (2) can transceiver; (3) battery pack; (4) power distribution board; (5) inertial measurement unit; (6) single trailing suspension; (7) hub motor; (8) extension motor & encoder; (9) thigh motor & encoder; (10) calf motor & encoder; (11) extension joint; (12) thigh; (13) calf; (14) foot.

To address these gaps, we propose a policy transfer method that incorporates both kinematic and dynamic constraints. Specifically, our Consistency-Aware Retargeting (CAR) method extends inverse kinematics. It enforces dynamic consistency to handle body-leg ratios and mass differences in heterogeneous transfers. As a result, CAR generates high-quality datasets, reducing joint overload and instability. Additionally, we integrate Bayesian Optimization (BO) into the training process. This automates searches for policy model structures and hyperparameters, boosting learning efficiency and stability. This research aims to achieve reliable policy transfer from Unitree Go1 to Tiangou. It offers a pathway for intelligent control in heavy-duty robots.

The main contributions of this study are as follows:

- We propose a CAR method that extends inverse kinematics with dynamic consistency constraints, including warm-start initialization, contact consistency, hybrid *Z*-mapping, root smoothing, and joint low-pass filtering.
- We integrate BO with Thompson sampling into the Proximal Policy Optimization (PPO) framework, enabling automated tuning of network architectures and hyperparameters.
- We demonstrate practical policy transfer from the lightweight Unitree Go1 to our self-developed heavy wheel-legged robot Tiangou (80 kg self-weight), providing a scalable foundation for heterogeneous robotic platforms in applications.

The paper is structured as follows: Section II reviews related work; Section III details the framework and formulations; Section IV presents experimental results and discussions. Section V provides the conclusion.

## II. Literature Review

In robotic policy transfer [11], motion retargeting and cross-platform adaptation address morphological differences across heterogeneous platforms. Existing studies can be categorized into three main areas: kinematics-based retargeting methods [12], unified learning frameworks for cross-morphology generalization [13], and imitation learning enhanced by Reinforcement Learning (RL) [14].

Motion retargeting captures key marker sequences and applies inverse kinematics for zero-shot transfer from source to target. For instance, Yoon *et al*. [15] proposed Spatio-Temporal Motion Retargeting (STMR). This generates executable sequences via inverse kinematics, efficiently reproducing animal motions on quadruped robots. Similarly, Fuchioka *et al*. [16] introduced OPT-Mimic, which optimizes trajectories to reduce noise. However, these methods focus on single-platform reproduction. Building upon this, policy transfer research has advanced toward constructing unified learning frameworks that enable multi-morphology adaptation through a single network. Liu *et al*. [17] presented the unified locomotion transformer. It achieves zero-shot generalization across diverse robot morphologies. Qin *et al*. [18] improved task-switching robustness via language-conditioned control. Yet, these methods require large datasets and high computation, and overlook dynamic discrepancies. Methods by Reske *et al*. [19] and Niu *et al*. [20] succeed on specific hardware. But they emphasize kinematics over nonlinear dynamics. This leads to failures when body proportions differ greatly. In particular, transfers from lightweight to heavy platforms cause oscillations and slippage. Another category optimizes imitation learning for better sample efficiency and generalization. Li *et al*. [21] proposed FastMimic. It integrates trajectory optimization with model-based controllers, imitating diverse gaits and minimizing hardware fine-tuning. Jin *et al*. [22] used staged objectives for simulation-to-reality transfer in high-speed running. Sood *et al*. [23] added a multi-critic RL framework. It balances imitation fidelity and task performance. This reduces reward tuning instability. Youm *et al*. [24] incorporated a fine-tuning method based on Model Predictive Control (MPC). Despite these advances, the RL-enhanced techniques have limitations. First, the PPO frameworks are sensitive to hyperparameters, making manual tuning labor-intensive. Second, relying solely on network augmentations cannot effectively address dynamic discrepancies across robot morphologies.

Overall, prior works have advanced motion mapping and learning optimization. However, they often overlook dynamic discrepancies in heterogeneous platforms and lack automated parameter tuning. Unlike kinematics-focused methods, which are limited to lightweight quadrupeds, our CAR method addresses mass and proportion mismatches in heavy-duty robots. Thus, we enhance robustness and efficiency for transfers from open-source to custom platforms via CAR and use BO for automated hyperparameter search.

## III. Methods

The proposed policy transfer framework is illustrated in Fig. 2. We generate source motion sequences from Go1's locomotion, which MPC controls. Based on the Inverse Kinematics (IK) of robots, we introduce mechanisms like time-consistent initialization, stance-phase foot locking, body smoothness constraints, and joint low-pass filtering. These form the CAR method, which creates a motion dataset for Tiangou. In the imitation module, we define multiple candidate hyperparameter configurations. Gaussian Process (GP) modeling with Thompson sampling is employed to explore the training pipeline adaptively. Finally, the module outputs joint commands executed on Tiangou via a PD controller.

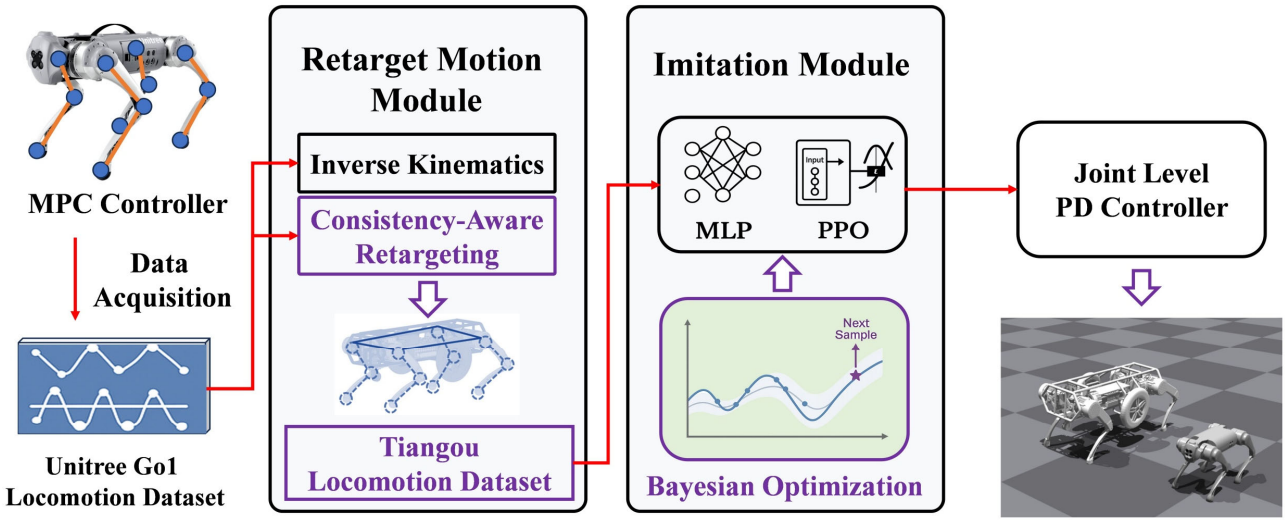


Fig. 2. Policy transfer pipeline from Unitree Go1 to Tiangou.

### A. Problem Formulation

This study transfers locomotion policies via an imitation learning framework. In IL, the policy drives the imitator to replicate the expert's behavior. The quality of the source locomotion sequences sets an upper Bound on the imitated policy's performance. Therefore, we select Go1's official MPC controller [25] as demonstrations $\mathcal{D}_{src}$, for its proven stability and gait diversity, ensuring high-fidelity datasets. Tiangou features distinct body and leg proportions, rendering direct imitation ineffective. Our objective is to learn a policy $\pi_\theta(a_t|o_t)$ that maps target observations $o_t \in \mathbb{R}^d$ to actions $a_t \in \mathbb{R}^m$, such that the policy should mimic source behaviors while adapting to target morphology. The optimization problem is shown in Eq. (1).

$$\theta^* = \arg\max_\theta \mathbb{E}_{\mathrm{T}\sim\pi_\theta}[\sum_{t=0}^{T}\gamma^t r(o_t, a_t)] \quad (1)$$

where $\gamma$ is the discount factor and $r(\cdot)$ is an imitation reward aligned with reference trajectories. The trained policy $\pi_\theta$ outputs joint angles as actions $a_t$, executed via a Proportional-Derivative (PD) controller with gains $K_P = 300$ and $K_D = 10$.

### B. Motion Retargeting

To address morphological differences, we map source trajectories $\{q_t^{src}\}$ to feasible target trajectories $\{q_t^{tar}\}$ using an IK-based pipeline. These compute the target base pose and foot positions by aligning key skeletal markers. We define 19 markers: the joints of the four legs, the center, the head, and the tail. The root pose is calculated in Eq. (2).

$$\begin{aligned} p_t^{root} &= \frac{1}{2}(p_{center} + p_{head}), \\ R_t^{root} &= Quat\left(f(p_{center}, p_{head}, p_{hips})\right) \end{aligned} \quad (2)$$

where $f(\cdot)$ constructs an orthogonal frame from center–head and hip vectors and $p_*$ the part position. End-effector targets are then mapped by geometric transfer in Eq. (3).

$$p_{i,t}^{toe,tar} = p_i^{hip,tar} + (p_i^{toe,src} - p_i^{hip,src}) + \Delta_{offset,i} \quad (3)$$

where $\Delta_{offset,i}$ donates the geometric difference of the source and target robots.

To evaluate the performance of the motion retargeting method, we define three metrics: $L2$ norm of foot position error (Eq. (4)), joint angular velocity $L2$ norm (Eq. (5)), and foot slippage $L2$ norm (Eq. (6)).

$$EE\_Pos\_L2 = \frac{1}{N}\sum_{i=1}^{N}\left\|p_i^{fk} - p_i^{tar}\right\|_2 \quad (4)$$

where $p_i^{fk}$ is the forward kinematics foot position and $p_i^{tar}$ the target foot position. Lower values show precise foot execution.

$$Joint\_Vel\_L2 = \|(q_t - q_{t-1})/\Delta t\|_2 \quad (5)$$

where $q_t$ is the joint angle vector and $\Delta t$ the interval. Lower values indicate smoother trajectories.

$$Foot\_Slip = \sum_{j\in\mathcal{C}} \left\| \left(p_{j,t}^{fk} - p_{j,t-1}^{fk}\right)_{xy} \right\|_2 \quad (6)$$

where $\mathcal{C}$ is the contact set and $(\cdot)_{xy}$, the horizontal projection. Lower values reduce unwanted slippage.

### C. Consistency-Aware Retargeting Components

Observations indicate that instability arises from mismatches in stance contact patterns, root placement, and joint feasibility. Thus, we add five enhancements to the IK backbone for CAR. The coefficients were empirically tuned based on the following considerations: smaller values for pose and joint filtering ensure a quick response and tracking. At the same time, the contact threshold is determined by foot geometry and simulation precision.

#### 1) Warm-start IK

This component initializes the IK solver with the solution from the previous frame as an anchor, leveraging temporal continuity to improve trajectory smoothness and significantly improve solving efficiency. Thereby, it can reduce foot slip and joint velocity spikes. The joint solution at frame $t$ is defined as $q_t = IK(p_{tar}^t, q_{rest})$. Where $p_{tar}^t$ denotes the target key marker position and $q_{rest}$ the rest pose. $q_{rest}$ is equal to $q_{t-1}$ $(t > 0)$ otherwise $q_{default}$, where $q_{default}$ means the default rest pose.

#### 2) Contact consistency

Independent solving per frame of IK causes cumulative foot position deviations, which worsen as the differences increase. We use contact consistency to lock the feet's horizontal positions upon contact, thereby reducing the cumulative slippage errors. This is an empirical constraint that enhances motion quality at the kinematic level. For end-effector $i$ (e.g., each foot), the stance phrase is deemed when the foot height $z < \epsilon$, where $\epsilon$ is set to 0.03 m. The target *XY*-positions are defined in Eq. (7).

$$[x_{tar,t}^i, y_{tar,t}^i]^T = \begin{cases} [x_{t-1}^i, y_{t-1}^i]^T, z < \epsilon \\ [x_{init}^i, y_{init}^i]^T, otherwise \end{cases} \quad (7)$$

where $x_{t-1}^i$, $y_{t-1}^i$ denote the previous frame positions and $x_{init}^i$, $y_{init}^i$ the initial positions.

#### 3) Hybrid z-mapping

Similar to contact consistency, this component locks the feet's vertical positions upon contact. For each foot $i$, the target *Z*-positions $z_{tar}^i$ and the relative *Z*-increment $\Delta z^i$ are defined in Eq. (8).

$$z_{tar}^i = \begin{cases} z_{init}^i + \Delta z^i, z < \epsilon \\ z_{src}^i, otherwise \end{cases} \quad (8)$$
$$\Delta z^i = z_{src}^i - z_{src}^{i,hip}$$

where $z_{tar}^i$, $z_{init}^i$, $z_{src}^i$ denote the target *Z*-position, the initial *Z*-position, and the reference *Z*-position in source data of foot $i$, respectively. $z_{src}^{i,hip}$ means the reference hip *Z*-position corresponding to the foot.

#### 4) Root smoothing

Motion retargeting causes high-frequency jitter in the generated data due to the noise in the reference data, structural differences between robots, and sensitivity in geometric calculations. To overcome the abrupt changes in the body, this component uses temporal smoothing filters. It weights the current frame's raw pose with the previous smoothed value. We set $\alpha$ = 0.18 for position averaging and $\beta$ = 0.2 for orientation interpolation. Therefore, root smoothing retains motion trends while adding inertia and delay, as defined in Eq. (9).

$$\begin{aligned} p_t^{root} &= (1-\alpha)p_{t-1}^{root} + \alpha p_t^{root} \\ R_t^{root} &= R_{t-1}^{root}\frac{\sin((1-\beta)\Omega)}{\sin(\Omega)} + R_t^{root}\frac{\sin(\beta\Omega)}{\sin(\Omega)} \\ \Omega &= \operatorname{acos}\left(R_{t-1}^{root}\cdot R_t^{root}\right) \end{aligned} \quad (9)$$

where $p_t^{root}$ denotes the world position of the root at time $t$, and $R_t^{root}$ denotes the orientation.

#### 5) Joint low-pass filter

Similar to root smoothing, this component performs temporal smoothing in joint space. It avoids high-frequency jitter in joint trajectories. A first-order exponential moving average low-pass filter is employed, with the factor $\eta = 0.12$ and updates recursively as shown in Eq. (10). Additionally, soft clipping is added to limit the joint Bounds, ensuring safer and biomechanically feasible motions.

$$q_t^{tar} \leftarrow (1-\eta)q_{t-1}^{tar} + \eta q_t^{src} \quad (10)$$

where $q_t^{tar}$ denotes the filtered target joint position at time $t$, $q_t^{src}$ the original value computed via the IK solver.

In summary, Warm-start IK addresses the root cause of abrupt trajectory changes in generated data. Contact consistency and hybrid *Z*-mapping impose dynamic constraints on horizontal and vertical directions to limit foot-end slippage. Root smoothing and joint low-pass filter suppress high-frequency jitter in root and joints, ensuring smooth motion. The complete CAR processing pipeline is shown in Algorithm 1.

**Algorithm 1: Motion Retarget Process via Car**

Initialize: prev_joints, prev_toe_fk, prev_root
1. for each frame t:
2. Compute original root pose.
3. Apply root smoothing
4. for each foot i
5. Compute the foot target.
6. Apply hybrid z-mapping
7. Apply contact consistency
8. Solve warm-start IK
9. Apply a joint low-pass filter.
10. Clip to limits
11. Assemble pose
12. Update states

### D. Learning with Bayesian Optimization

Using retargeted demonstrations, we apply imitation learning within an actor–critic framework [26]. The policy

is a Multi-Layer Perceptron (MLP) optimized via PPO. The surrogate loss is shown in Eq. (11).

$$L(\theta) = \mathbb{E}_t[\min(h_t(\theta)\hat{A}_t, clip(h_t(\theta), 1 \mp \epsilon)\hat{A}_t)] \\ h_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{old}}(a_t|s_t) \quad (11)$$

where $h_t(\theta)$ donates the difference of two policies and $\hat{A}_t$ the Generalized Advantage Estimation (GAE), $\epsilon$ is the threshold constant.

The effectiveness of policy depends critically on hyperparameters (e.g., learning rate, clipping ratio, batch size) and network architecture (e.g., depth, activation functions). We use a discrete search space for BO to improve computational efficiency and reduce overfitting risks. This choice limits candidates to practical values, avoiding excessive evaluations in continuous spaces [27].

Parameter ranges extend from baseline values to cover conservative and aggressive options. For network architecture, we draw from OPT-Mimic's standard settings, with sizes ranging from 256 (ensuring sufficient expression capacity) to 1024 (avoiding overfitting), and common RL activations (Tanh, Rectified Linear Unit (ReLU), Exponential Linear Unit (ELU)) [16]. For PPO hyperparameters, ranges balance stability and exploration, such as learning rates (5e−4, 1e−5, 5e−5, 1e−4). We employ the Radial Basis Function (RBF) kernel due to its low computational complexity and suitability for discrete jumps in candidates, such as activation types, where continuous similarity is lacking. The RBF defines the covariance matrix as $k(h, h') = \exp(-\|h - h'\|^2/(2l^2))$, with $l$ as length scale. Each candidate $h$ follows a Gaussian posterior $\mathcal{N}(\mu_h, \sigma_h^2)$. Thompson sampling selects the next $h$ by sampling from this posterior, favoring areas of high uncertainty or high mean values. The posterior updates via Bayesian conjugate, depicted in Eq. (12), for the new noisy average return $y$ with variance $\sigma_n^2$.

$$\mu_{h,new} = \mu_h + k(\sigma_h^2 + \sigma_n^2)^{-1}(y - \mu_h) \\ \sigma_{h,new}^2 = \sigma_h^2(1 - k) \\ k = \sigma_h^2(\sigma_h^2 + \sigma_n^2)^{-1} \quad (12)$$

where $k$ is the kernel vector between $h$ and the observed points. $\mu_h$ represents the expected performance level of the robot under configuration $h$. $\sigma_h^2$ quantifies the uncertainty in this performance, enabling Thompson sampling to prioritize high-uncertainty or high-mean areas for efficient optimization.

We adopt the reward formulation of GenLoco [10], which establishes rewards based on body position, velocity, and end-effector errors with scale normalization. These terms ensure stable locomotion by aligning behaviors. Scale normalization adjusts error tolerance globally. We follow this design but fine-tune it for Tiangou. The reward $r_t$ is given by Eq. (13).

$$r_t = w^p r_t^p + w^v r_t^v + w^{bp} r_t^{bp} + w^{bv} r_t^{bv} \quad (13)$$

where $r_t^p$ and $r_t^v$ encourage Tiangou to match the joint angles and angular velocities of the reference dataset; the reward $r_t^{bp}$ and $r_t^{bv}$ serve as corresponding body-level constraints. Weights $w^*$ prioritize dynamics: $w^p = 0.6$ emphasizes joint configurations' matching to the reference due to high inertia, as the highest priority; $w^v = 0.1$ favors motion smoothness over velocity tracking errors; $w^{bp} = 0.15$ and $w^{bv} = 0.15$ ensure root attitude stability.

The overall algorithm is summarized in Algorithm 2, presented as pseudo-code.

**Algorithm 2: Policy Transfer with Consistency-Aware Retargeting and Bayesian Optimization**

Input: Source dataset $\mathcal{D}_{src}$, target robot model, number of trials $N$, training steps $K$ per trial, noise variance $\sigma_n^2$
Output: Optimized imitation policy $\pi_\theta^*$

1. Retarget source motions using CAR → $\mathcal{D}_{tar}$
2. Initialize Thompson sampler with Gaussian priors of each hyperparameter $h$ (e.g., $N(\mu_h = 0, \sigma_h^2 = 100)$)
3. For trial = $1, \ldots, N$ do
   // *Thompson Sampling*
   for each candidate $h_i$ do
   Sample from posterior: $s_i \sim N(\mu_h, \sigma_h^2)$)
   end for
   Select $h$ with maximum $s_i$
   // *Train PPO with selected h for K steps on* $\mathcal{D}_{tar}$
   Initialize robot state
   for step =1 to $K$ do
4. Observe the current state $o_t$
   Compute action $a_t = \pi_\theta(a_t|o_t)$
   Execute action, update state
   end for
   // *Test policy performance over episodes*
   Initialize evaluation episodes
   for each episode do
   Start with initial observation
5. Run policy for steps until termination, accumulate reward
   end for
   Compute the average return $y$
   // *Update posterior for selected h*
   Compute Kalman gain $k$
6. Update mean $\mu_{h,new}$
   Update variance $\sigma_{h,new}^2$
7. End For
8. Select $h^*$ with the highest posterior mean $\mu_h$
9. Retrain PPO policy $\pi_\theta$ with $h^*$ until convergence
10. Return $\pi_\theta^*$

## IV. Results and Discussion

Using the proposed framework, we controlled Unitree Go1 to perform six gaits—Pace, Trot, Bound, Canter, Turn Left, and Turn Right—sequentially on flat ground, thereby collecting datasets. Ablation studies were followed by subsequent real-world deployment to assess the effectiveness of policy transfer.

### A. Ablation Studies

In the Gym environment, we applied IK and CAR retargeting to Go1's six gaits and projected them onto Tiangou. All metrics represent cumulative values aggregated across all four legs and the entire motion sequence, as defined in Eqs. (6)–(8). CAR consistently

reduced foot-end position errors across all gaits. For instance, errors dropped from 0.197 m to 0.108 m in Pace and from 0.17 m to 0.054 m in Turn Right, achieving a 37% average reduction. Ablation results show that removing Contact Consistency (CC = False) causes a significant increase in error, up to 138% in dynamic gaits, such as Trot. Other removals, such as Root Smoothing (RS = False), add less than 1% errors on average, as shown in Fig. 3. Joint angular velocity spikes were suppressed under CAR. Examples included a decline from 22.92 rad/s to 12.55 rad/s in Canter and from 9.9 rad/s to 4.4 rad/s in Pace, corresponding to a 45% average improvement. Here, the Joint_Vel_L2 metric (rad/s) represents the $L2$ norm of angular velocity changes, capturing both average fluctuations and peak spikes over the sequence. Removing joint low-pass filtering (JL = False) leads to the highest velocity spikes, increasing by 131% in Trot and 84% in Canter. Warm-Start (WS = False) and hybrid $Z$-mapping (HZ = False) contribute less than 10% to smoothness in stance-heavy gaits, as shown in Fig. 4. Undesired stance-phase slippage diminished markedly, such as from 66.5 mm to 25.3 mm in Turn Left and from 25 mm to 4.72 mm in Pace. Contact consistency removal (CC = False) degrades slippage most, by 101% in Pace and 71% in Canter. Root Smoothing (RS = False) introduces 15% slippage in rotations, confirming synergies among enhancements, as shown in Fig. 5. Notably, CAR incurs no significant computational overhead, as evidenced by comparable processing times in Fig. 6.

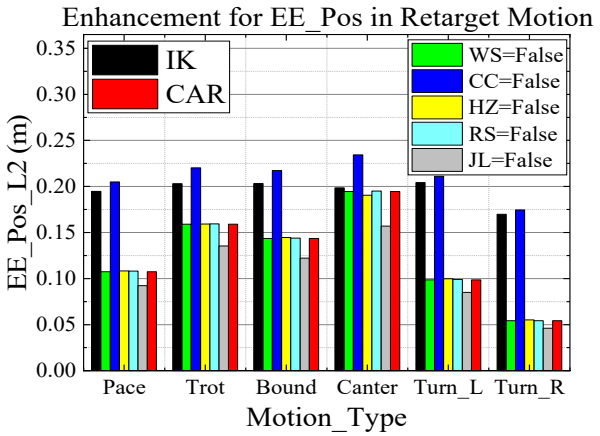


Fig. 3. Comparison of end-effector position errors.

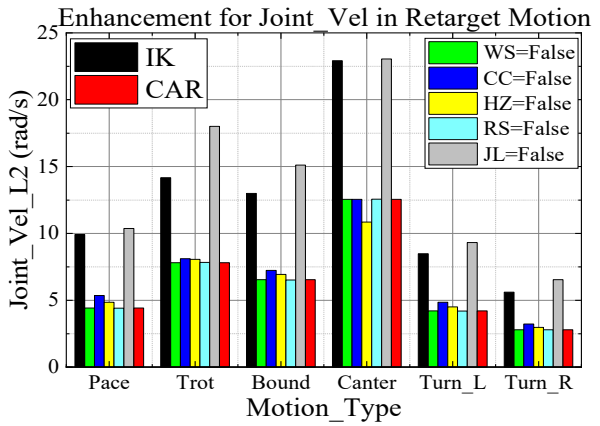


Fig. 4. Comparison of joint angular velocity smoothness.

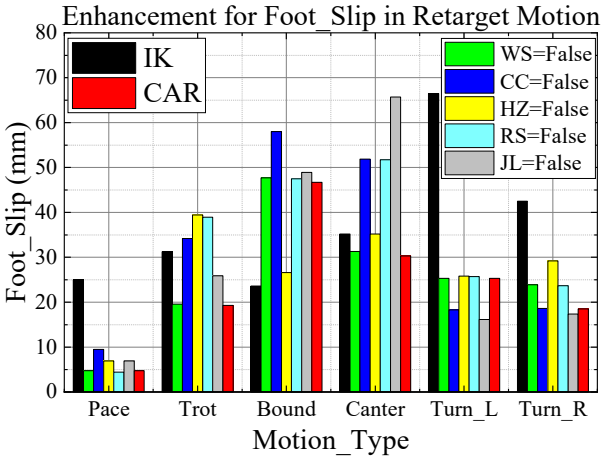


Fig. 5. Comparison of foot slip rates.

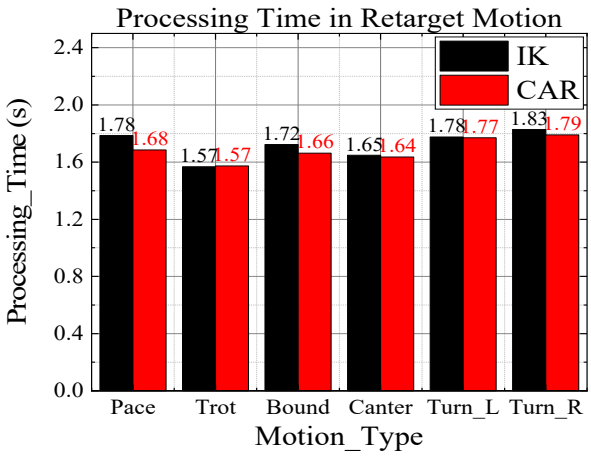


Fig. 6. Comparison of motion retargeting processing time.

## B. *Policy Training*

To evaluate BO's role in the imitation module, we utilized CAR-generated reference data and expanded the baseline policy with combinations of activation functions and network sizes, yielding 21 candidate configurations (Table I). Similarly, we defined six combinations of sensitive PPO hyperparameters, including learning rates (Table II). The search space encompassed 64,512 combinations, finally.

Due to computational constraints, we adopted a hierarchical approach for BO optimization instead of exhaustively traversing the whole 64,512-combination space. Ultimately, this process resulted in a pool of 28 configurations. Fig. 7 illustrates the BO convergence over 40 trials in the final 28 candidates. The blue line shows per-trial scores, while the red dashed line tracks the best-so-far cumulative optimum. The highest score, 5.748, occurred at trial 12, corresponding to the optimal combination detailed in Table III. The sampling frequency of the optimization process for the final 28 candidates is presented in a heatmap, as shown in Fig. 8. The rows represent hyperparameter triplets, and the columns denote activation types. Hotspots cluster around ReLU with small learning rates (on the order of 1e−5), moderate clip values (0.03–0.1), and batch sizes (64–128), reflecting adaptive sampling based on surrogate model predictions.

TABLE I. CANDIDATE OPTIMIZATION SPACE OF NETWORK ARCHITECTURE PARAMETERS

| Parameter | Baseline | Optimization Candidates |
|---|---|---|
| Activation Function | ReLU | [Tanh, ReLU, Elu] |
| Actor Network | [512, 512] | [256, 256], [512, 256], [256, 512], [512, 512], [1024, 512], [512, 1024], [1024, 1024] |
| Critic Network | [512, 512] | [256, 256], [512, 256], [256, 512], [512, 512], [1024, 512], [512, 1024], [1024, 1024] |

TABLE II. CANDIDATE OPTIMIZATION SPACE OF PPO HYPERPARAMETERS

| Parameter | Baseline | Optimization Candidates |
|---|---|---|
| Learning Rate | 1e−5 | [5e−4, 1e−5, 5e−5, 1e−4] |
| Clipping Parameter | 0.2 | [0.1, 0.2, 0.3, 0.4] |
| Optimization Epoch | 1 | [1, 2, 3] |
| Batch Size | 64 | [32, 64, 128, 256] |
| Discount Factor | 0.95 | [0.93, 0.94, 0.95, 0.96] |
| GAE $\lambda$ | 0.95 | [0.93, 0.94, 0.95, 0.96] |

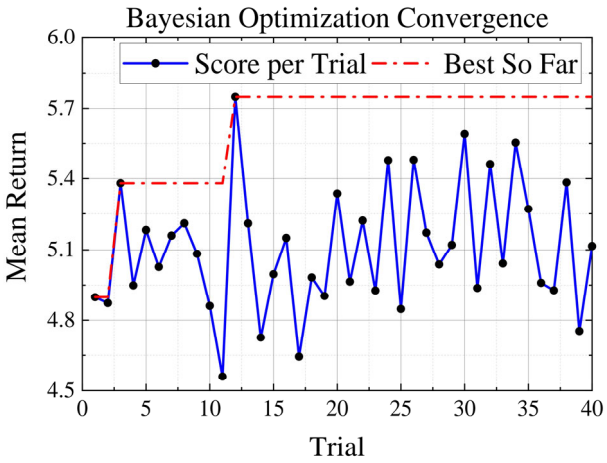


Fig. 7. BO convergence of the final optimization.

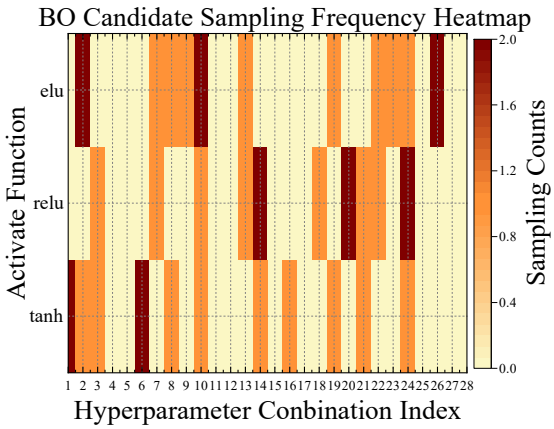


Fig. 8. Sampling frequency heatmap of the final optimization.

TABLE III THE OPTIMAL PARAMETERS YIELDED VIA BAYESIAN OPTIMIZATION

| Parameter | Optimal | Parameter | Optimal |
|---|---|---|---|
| Activation Function | ReLU | Learning Rate | 1e−5 |
| Actor Network | [512, 256] | Batch Size | 128 |
| Critic Network | [512, 256] | Discount Factor | 0.95 |
| Clipping Parameter | 0.1 | Optimization Epoch | 1 |
| GAE $\lambda$ | 0.95 | - | - |

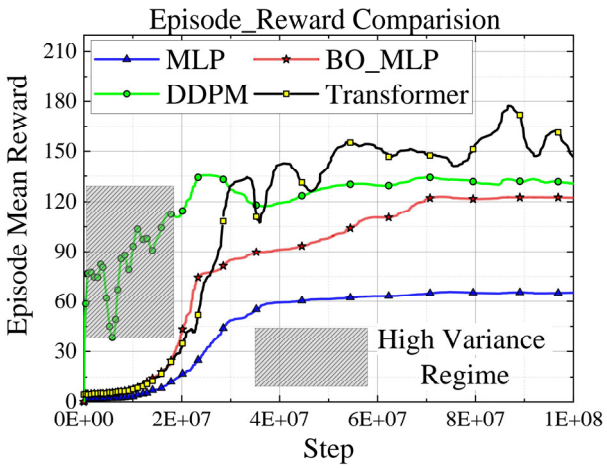


Fig. 9. Comparison of episodic rewards.

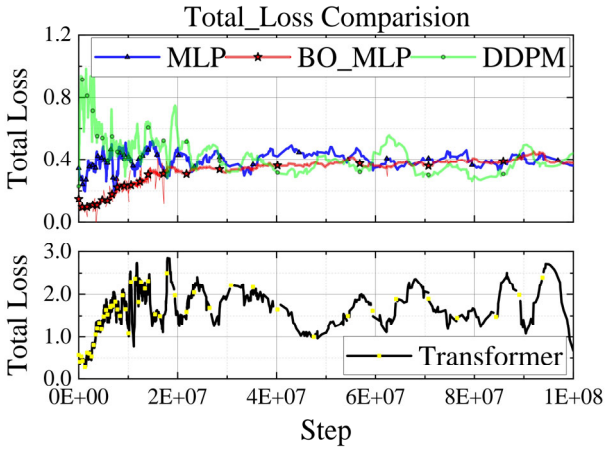


Fig. 10. Comparison of total loss.

We performed a grid search using Thompson Sampling and limited the BO process to 1E5 steps. To demonstrate the superiority of our BO strategy in enhancing training stability and efficiency, we further constructed Actor-Critic training architectures based on Denoising Diffusion Probabilistic Model (DDPM) and Transformer. DDPM was selected for its probabilistic generative nature, which excels in modeling complex distributions but often introduces early variance in robotics tasks. The Transformer was chosen due to its attention mechanisms, which enable effective sequence handling while being sensitive to hyperparameters in policy learning. In contrast to the baseline, which converged at around 2E7 steps with an episode reward of 65, the BO-augmented policy surpassed 70 at the same point and stabilized above 120 after 6E7 steps. The DDPM policy showed initial oscillations but reached 100 after 4E7 steps. The Transformer-Based policy achieved over 140, but with frequent fluctuations, as shown in Fig. 9. Moreover, total loss fluctuations were minimized under BO, dropping

rapidly below 0.1 and stabilizing between 0.3 and 0.4. The DDPM loss oscillates rapidly between 0.4 and 1.0 before 2E7 steps, and finally decreases to around 0.5. In contrast, the loss of the Transformer fluctuates continuously between 1.0 and 3.0, as shown in Fig. 10.

### C. Real-World Experiment

We deployed the trained policy on Tiangou's central controller, while Go1 retained its MPC controller from the data collection phase. Both robots utilized the Lightweight Communications and Marshalling (LCM) protocol, which included noise offsets from differing encoders. The Inertial Measurement Unit (IMU) was mitigated via low-pass filtering. Joystick commands—"forward–backward–turn–stand"—enabled comparisons of behavioral differences of the two robots. We then evaluated Tiangou's mobility and load-bearing capabilities, as illustrated in Fig. 11.

Fig. 11. Real-world testing of the transferred policy.

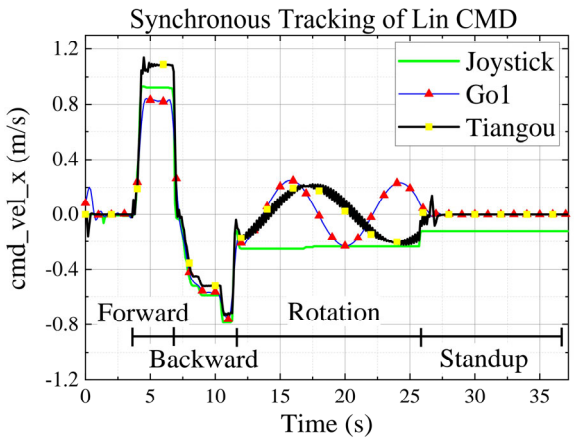


Fig. 12. Forward command tracking ability.

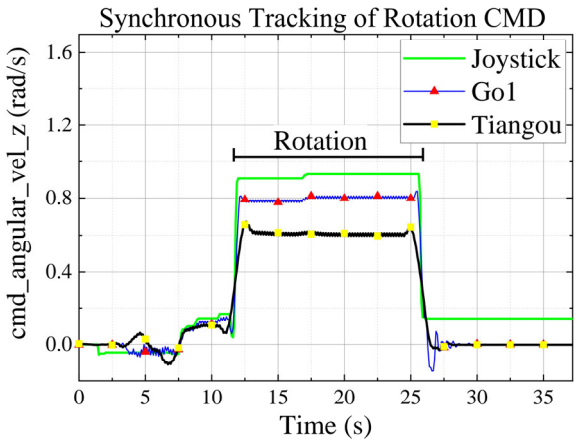


Fig. 13. Turning command tracking ability.

Under a 0.9 m/s forward and −0.8 m/s backward command, Go1 tracked ±0.8 m/s, whereas Tiangou achieved 1.1 m/s forward and −0.8 m/s backward (Fig. 12). For steering at 1.0 rad/s, Go1 reached 0.8 rad/s, but Tiangou stabilized at 0.6–0.7 rad/s, requiring a forward velocity component to avoid anomalies Fig. 13.

Furthermore, FR-Hip joint torque peaked at 5 Nm for Tiangou versus 2 Nm for Go1, with both maintaining stable angles and velocities (Fig. 14). The body orientation curves remained consistent. However, Tiangou's larger size reduced yaw agility (Fig. 15).

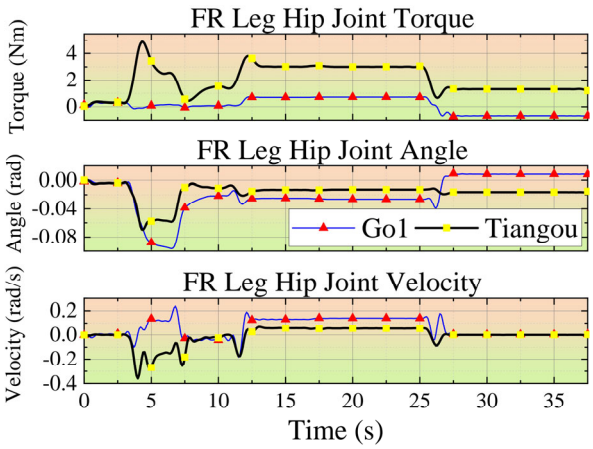


Fig. 14. Dynamic characteristics of FR-Hip during locomotion.

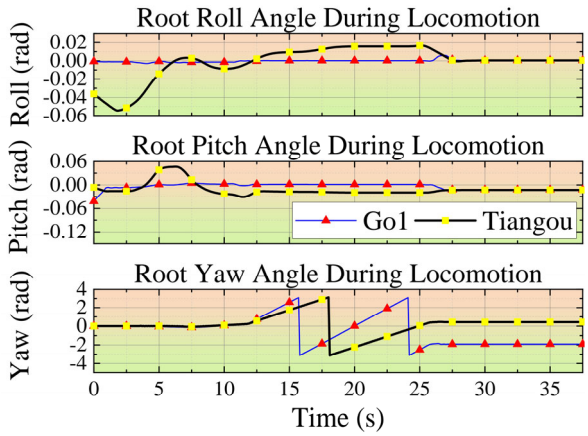


Fig. 15. Body behavior during locomotion.

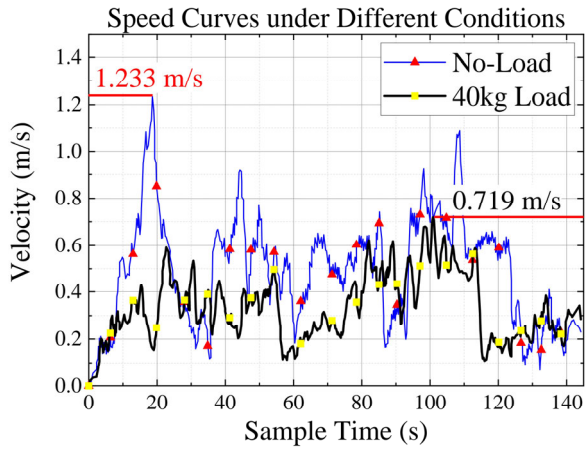


Fig. 16. Tiangou mobility test.

In Tiangou's capability tests, the no-load maximum speed reached 1.233 m/s, with averages ranging from 0.6 to 0.8 m/s. Under a 40 kg payload, maximum speed fell to 0.719 m/s. Averages shifted to 0.2–0.6 m/s, as shown in Fig. 16. During initial loading, body height dipped from 0.50 m to 0.39 m before recovering to 0.42 m; turning phases showed stable height with yaw varying smoothly from 0 to over 5 rad, as shown in Fig. 17.

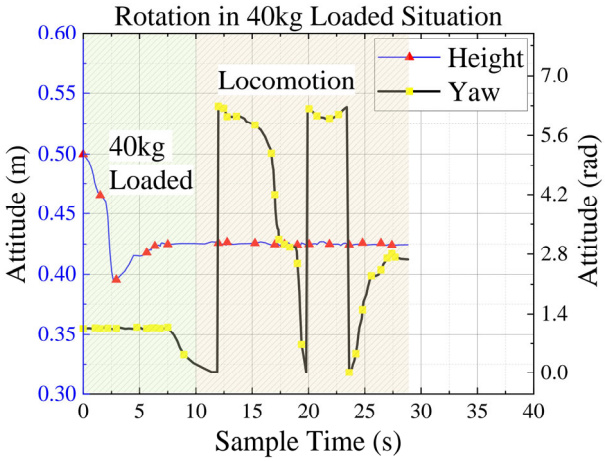


Fig. 17. Tiangou turning performance test.

### D. Discussion

The experimental results demonstrate that the CAR method, by incorporating dynamic consistency constraints into inverse kinematics, effectively reduces foot-end position errors, joint angular velocity fluctuations, and foot-end slippage while preserving computational efficiency. This enhancement stems from mechanisms such as contact consistency, which ensures that projected trajectories adhere to physical laws and mitigate dynamic discrepancies between heterogeneous robots. In particular, CAR generates higher-fidelity imitation datasets for platforms with significant morphological differences, providing a robust basis for PPO-based training. Integrating BO further boosts cumulative rewards and training stability, with optimized policies nearly doubling reward values and reducing loss fluctuations. Mechanistically, BO's use of Thompson sampling enables efficient exploration of the hyperparameter space, thereby avoiding the overhead associated with manual or grid-based searches. This automated approach enhances the robustness of imitation learning in reinforcement learning contexts.

Compared to prior policy transfer models, such as GenLoco [10], which rely on kinematics-based scaling for quadrupeds of similar size, CAR excels in heterogeneous settings. Quantitatively, CAR reduces foot-end position errors by 40% on average versus the IK baseline (inspired by GenLoco's retargeting), as shown in Fig. 3. It improves joint velocity smoothness by 45% (Fig. 4) and decreases foot slippage (Fig. 5). In imitation learning, our BO-augmented PPO outperforms standard MLP baselines, such as in FastMimic [21], by doubling cumulative rewards (from 65 to over 120, Fig. 9) and stabilizing loss below 0.4 (Fig. 10). The DDPM baseline exhibited early oscillations in reward curves due to its stochastic denoising process, which introduces variance in initial trajectory sampling. DDPM loss values generally met expectations, settling at 0.5. However, the major drawback lies in its low computational efficiency, resulting in training times that are 15 times longer compared to MLP and BO-MLP. The Transformer achieved higher rewards over 140 but showed frequent curve fluctuations and larger losses above 1.0 with evident overshoots. Transformer-based networks require more fine-tuned hyperparameters to mitigate the risk of overfitting. Future work on optimizing Transformer parameters could be an interesting direction. Qualitatively, CAR enforces contact and smoothness constraints, unlike kinematics-only methods that ignore dynamic mismatches. This enables stable transfer to heavy platforms, such as Tiangou. Furthermore, our method advances beyond kinematics-focused retargeting by achieving more stable velocity tracking and posture control in real-world transfers from Go1 to Tiangou. Although Tiangou showed slight latency in turning (0.6–0.7 rad/s response versus Go1's 0.8 rad/s), trajectories remained smooth without overshoot, consistent with its higher inertia. Load tests under 40 kg confirmed policy robustness: the peak velocity dropped by about 40%, but posture recovery was rapid, and yaw variations remained minimal, indicating adaptability to disturbances. These outcomes underscore the engineering value of dynamic constraints for heavy-legged robots.

Despite these advancements, CAR shows limitations in extreme scenarios. For instance, limited leg lift height prevents crossing 15 cm steps. Compared to our direct dynamics-based control of Tiangou, this study reduces energy efficiency in legged mode, lasting only 40 min versus 2.5 h. In experiments, Tiangou handles load variations from 0 to 40 kg; beyond this range, the actuators overheat rapidly. Sudden acceleration commands cause falls, indicating poor response to abrupt inputs. These failure cases highlight CAR's constrained generalization to varied terrain, heavy payloads, and dynamic commands. Additionally, BO's sampling efficiency diminishes in high-dimensional spaces, potentially requiring more iterations for complex sets. Future work could integrate adaptive dynamics modeling to enhance robustness.

Future work could address these by extending evaluations to diverse platforms and scenarios, incorporating meta-learning for faster adaptation, and integrating energy optimization to improve deployment efficiency. Overall, this study highlights the integration of dynamics-aware retargeting and automated optimization as a foundation for policy transfer across heterogeneous systems, with implications for industrial applications that require heavy loads.

## V. Conclusion

This study introduces the CAR method, incorporating dynamic consistency constraints on top of inverse kinematics, and integrates it with Bayesian optimization within a PPO framework. This enables effective policy transfer from Unitree Go1 to the heavy wheel-legged

Tiangou. Experimental results validated that CAR achieved an average 40% reduction in foot-end position errors, 45% improvement in joint angular velocity smoothness, and substantial decreases in foot-end slippage. BO nearly doubled cumulative rewards from 65 to over 120 and enhanced training stability. Thus, our approach outperforms kinematics-only methods by ensuring robustness across significant morphological differences, including a 7.7× mass disparity. Although limited to flat terrain and a single platform, this work establishes a novel paradigm for transferring heterogeneous robot policies. It demonstrates practical feasibility for heavy-duty applications in disaster response and industrial operations. Future directions prioritize extending validations to multiple platforms for cross-morphology generalization, incorporating meta-learning for rapid adaptation, and adding energy optimization for enhanced efficiency. Overall, these advancements pave the way for scalable imitation learning in robotics, fostering broader deployment in real-world scenarios.

## Conflict of Interest

The authors declare no conflict of interest.

## Author Contributions

CH: Conceptualization, Methodology, Visualization, On-field Formal analysis, Investigation, Writing; LX: Resources, Project administration, Supervision, Funding acquisition; CHu: Test setup, Measurements, Visualization; all authors had approved the final version.

## Funding

This research was funded by the China Hubei Province Key R&D Program (Grant No. 2020DEB014).

## Acknowledgment

The authors wish to thank Hubei Key Laboratory of Advanced Technology for Automotive Components and New Energy & Intelligent Vehicle DreamWorks of Wuhan University of Technology for the continuous support.

## References

[1] A. M. El-Dalatony, T. Attia, H. Ragheb *et al*., "Cascaded pid trajectory tracking control for quadruped robotic leg," *Int. J. Mech. Eng. Robot. Res*., vol. 12, no. 1, pp. 40–47, 2023.

[2] V. T. Hà and T. T. Thương, "Neural-backstepping adaptive control for nonlinear motion of sliding mobile robots," *Int. J. Mech. Eng. Robot. Res.,* vol. 14, no. 3, pp. 323–339, 2025.

[3] C. Wang, C. Li, Q. Han *et al*., "A performance analysis of a litchi picking robot system for actively removing obstructions, using an artificial intelligence algorithm," *Agronomy*, vol. 13, no. 11, 2795, 2023.

[4] R. Edlinger, C. Föls, and A. Nüchter, "An innovative pick-up and transport robot system for casualty evacuation," in *Proc. 2022 IEEE Int. Symp. Safety, Security, Rescue Robot. (SSRR)*, Sevilla, 2022, pp. 67–73.

[5] T. Sellers, T. Lei, H. Rogers *et al*., "Autonomous multi-robot allocation and formation control for remote sensing in environmental exploration," in *Proc. SPIE 12540, Unmanned Systems Technology XXV*, Orlando, 2023, 125400U.

[6] I. V. Merkuryev, T. B. Duishenaliev, G. Wu *et al*., "Robust control of a 2D rehabilitation robot using admittance and RBF neural network," *Int. J. Mech. Eng. Robot. Res.,* vol. 14, no. 5, pp. 511–524, 2025.

[7] X. Feng, S. Liu, Q. Yuan *et al*., "Research on wheel-legged robot based on LQR and ADRC," *Sci. Rep.,* vol. 13, 15122, 2023.

[8] A. H. Abdulwahab, A. Z. A. Mazlan, A. F. Hawary *et al*., "Quadruped robots mechanism, structural design, energy, gait, stability, and actuators: A review study," *Int. J. Mech. Eng. Robot. Res.,* vol. 12, no. 6, pp. 385–395, 2023.

[9] D. Hoeller, N. Rudin, D. Stevšić *et al*., "ANYmal parkour: Learning agile navigation for quadrupedal robots," *Sci. Robot.,* vol. 9, no. 88, 2024.

[10] G. Feng, H. Zhang, Z. Li *et al*., "GenLoco: Generalized locomotion controllers for quadrupedal robots," in *Proc. 6th Conf. Robot Learning (CoRL)*, Auckland, 2022, pp. 1893–1903.

[11] M. Lucke, T. Hamadi, E. Rueckert *et al*., "Adaptation and transfer of robot motion policies for close proximity human-robot interaction," *Front. Robot. AI*, vol. 6, 69, 2019.

[12] Z. Chen, "A whole-body motion imitation framework from human data for full-size humanoid robot," arXiv preprint, arXiv: 2508.00362, 2025.

[13] S. Xiong, J. K. Gupta, C. Finn *et al*., "Universal morphology control via contextual modulation," in *Proc. 40th Int. Conf. Mach. Learn. (ICML)*, Honolulu, 2023, pp. 39376–39399.

[14] Z. Zhang, J. Liu, X. Wang *et al*., "Imitation-enhanced reinforcement learning with privileged smooth transition for hexapod locomotion," *IEEE Robot. Autom. Lett.*, vol. 10, no. 1, pp. 350–357, 2025.

[15] T. Yoon, S. Nam, J. Park *et al*., "Spatio-temporal motion retargeting for quadruped robots," arXiv preprint, arXiv: 2404.11557, 2024.

[16] Y. Fuchioka, Z. Xie, and M. van de Panne, "OPT-Mimic: Imitation of optimized trajectories for dynamic quadruped behaviors," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, London, 2023, pp. 1029–1035.

[17] D. Liu, J. Wu, Y. Ma *et al*., "Unified locomotion transformer with simultaneous sim-to-real transfer for quadrupeds," arXiv preprint, arXiv: 2503.08997, 2025.

[18] X. Qin, Z. Yuan, Z. Li *et al*., "Integrating diffusion-based multi-task learning with online reinforcement learning for robust quadruped robot control," arXiv preprint, arXiv: 2507.05674, 2025.

[19] A. Reske, J. Carius, Y. Ma *et al*., "Imitation learning from MPC for quadrupedal multi-gait control," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Xi'an, 2021, pp. 5014–5020.

[20] Y. Niu, Y. Zhang, M. Yu *et al*., "Human2LocoMan: Learning versatile quadrupedal manipulation with human pretraining," in *Proc. Robot.: Sci. Syst. (RSS)*, Los Angeles, 2025.

[21] T. Li, J. Won, J. Cho *et al*., "FastMimic: Model-based motion imitation for agile, diverse and generalizable quadrupedal locomotion," *Robotics*, vol. 12, no. 3, 90, 2023.

[22] Y. Jin, X. Liu, Y. Shao *et al*., "High-speed quadrupedal locomotion by imitation-relaxation reinforcement learning," *Nat. Mach. Intell.,* vol. 4, no. 12, pp. 1116–1128, 2022.

[23] S. Sood, C. Mavrogiannis, S. Srinivasa *et al*., "APEX: Action priors enable efficient exploration for skill imitation on articulated robots," arXiv preprint, arXiv: 2505.10022, 2025.

[24] D. Youm, H. Jung, H. Kim *et al*., "Imitating and finetuning model predictive control for robust and symmetric quadrupedal locomotion," *IEEE Robot. Autom. Lett.,* vol. 8, no. 11, pp. 7515–7522, 2023.

[25] Y. Chen and Q. Nguyen, "Learning agile locomotion and adaptive behaviors via RL-augmented MPC," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Yokohama, 2024, pp. 5984–5990.

[26] J. Wu, G. Xin, C. Qi *et al.*, "Learning robust and agile legged locomotion using adversarial motion priors," *IEEE Robot. Autom. Lett.,* vol. 8, no. 9, pp. 5575–5582, 2023.

[27] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in Neural Information Processing Systems,* 25, 2012.

Copyright © 2026 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0).