Efficient and Versatile Quadrupedal Skating: Optimal Co-design via Reinforcement Learning and Bayesian Optimization

Hanwen Wang[1,†], Zhenlong Fang[1,†], Josiah Hanna[1], Xiaobin Xiong[1,2]

***Abstract*— In this paper, we propose an effective mechanical and algorithmic solution to enabling skating motion with passive wheels on state-of-the-art quadrupedal robots. The skating locomotion enables a hybrid combination of wheeled and legged mobility without the necessity of motorization at the feet, which simultaneously promote efficiency, speed, and mechanical simplicity. To realize these potential advantage of skating, we employ a bilevel optimization approach with *an upper level optimization via Bayesian Optimization (BO)* to search for the best mechanical design and *a lower level Reinforcement Learning (RL)* to find an optimal motor policy. The end results not only provide optimal mechanical and control designs but also show versatile locomotion behaviors such as *hockey stop* (rapid braking by turning sideways to maximize friction) and *self-aligning* motion (automatic reorientation to maximize energy efficiency in the moving direction), providing the first system-level study on quadrupedal robotic skating.**

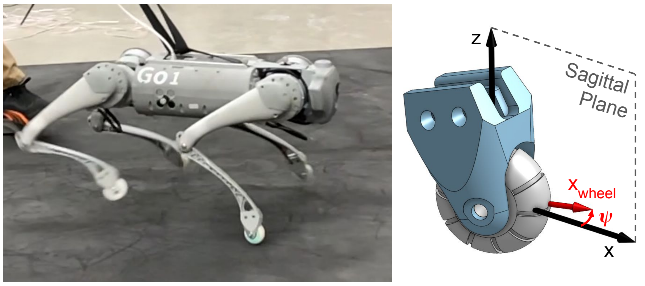

Fig. 1: Quadrupedal skating setup with passive wheels. Each foot is equipped with a 3D-printed roller support that holds a passive wheel. In the default standing configuration, the wheel yaw installation angle ψ is defined as the deviation of the wheel's x-axis, x_{wheel}, from the x-axis of the robot's sagittal plane. This angle is our key design parameter.

I. Introduction

Legged robots have made rapid progress in agility and robustness, enabling increasingly capable real-world deployment [1], [2], [3]. However, compared to wheeled platforms, legged locomotion still lags in speed and energetic efficiency [4], [5]. Wheels provide smooth contact and favorable energy scaling [4], but often trade off versatility on unstructured terrain. Hybrid systems such as wheeled-legged robots aim to bridge this gap [6], [7], but most existing designs rely on actively driven wheels, adding actuation, mass, and control complexity.

We explore an alternative hybrid mode—*quadrupedal skating with passive wheels*. Equipping each foot with a passive wheel allows a quadruped to retain the stability and maneuverability of legged locomotion while gaining the efficiency and speed of wheeled motion, without motorizing the feet. Figure 1 shows our implementation on a Unitree Go1 using custom 3D-printed roller supports with an adjustable wheel yaw installation angle ψ.

Achieving effective skating, however, is challenging. Skating dynamics couple robot posture and velocity under non-holonomic rolling constraints, requiring coordinated regulation of contact forces, body posture, and momentum exchange. With passive wheels, the robot cannot directly command wheel traction; instead, skating performance depends strongly on the interaction between mechanical design and the learned motor policy. This tight morphology-control coupling makes *co-design* essential: poor hardware configurations can restrict feasible behaviors, while suboptimal policies may fail to exploit an otherwise capable design.

To address this coupling, we propose a bilevel co-design framework that jointly optimizes mechanical parameters and a control policy. The upper level uses Bayesian Optimization (BO) [8] to efficiently search the design space, while the lower level uses Reinforcement Learning (RL) to train a policy specialized to each candidate design [9], [10]. This BO-RL loop discovers design-policy pairs that enable both efficient and versatile skating.

Our main contributions are as follows:

- **A novel locomotion paradigm:** We introduce quadrupedal skating with passive wheels, demonstrating its potential for combining speed, efficiency, and maneuverability.
- **A bilevel co-design framework:** We present a BO-RL approach that jointly optimizes wheel installation parameters and the motor control policy.
- **Versatile locomotion demonstrations:** We validate the method on a state-of-the-art quadruped and demonstrate behaviors such as *hockey stop* (rapid braking by turning sideways to maximize friction) and *self-aligning* motion (automatic reorientation to maximize energy efficiency in the moving direction).
- **A systematic study of quadrupedal skating:** We provide, to the best of our knowledge, the first system-level study of dynamic skating locomotion on quadrupedal robots.

Overall, our results suggest that co-designing morphology and control is a practical route to reliable quadrupedal skating

[†] Hanwen Wang and Zhenlong Fang contributed equally in this work.

[1] Hanwen Wang, Zhenlong Fang, and Josiah Hanna are with the University of Wisconsin - Madison. [2] Xiaobin Xiong was with the University of Wisconsin - Madison, and now with the Shanghai Innovation Institute (SII). Corresponding to Xiaobin Xiong: xiaobin.xiong@sii.edu.cn.

behaviors that are difficult to achieve with fixed, hand-designed configurations.

II. Related Work

A. Quadrupedal Locomotion

Quadrupedal robotics has progressed rapidly, with platforms demonstrating dynamic mobility and robust terrain traversal. Early hydraulic systems such as BigDog established rough-terrain feasibility [1]. Subsequent robots such as MIT Cheetah [11] highlighted the effectiveness of model-based control, including whole-body dynamics and model predictive control, for agile running and jumping [3]. More recently, Reinforcement Learning (RL) has produced highly dynamic locomotion skills and controllers that can match or exceed expert-designed strategies [2], [9], [12], [13], enabling robust deployment on systems such as ANYmal [14]. Despite these advances, pure legged locomotion is typically slower and less energy-efficient than wheeled motion, motivating wheeled–legged hybrids that combine both modalities [15], [16], [17], [18]. Many hybrid platforms, however, place substantial actuation and mass at the distal end-effector (e.g., driven wheels at the feet), increasing mechanical complexity and potentially reducing reliability.

B. Robot Skating

Skating locomotion has received comparatively limited attention, despite offering an appealing trade-off between efficiency and maneuverability on smooth terrain. A distinguishing feature is the coupling of propulsion and gliding: robots generate momentum during push-off phases and then sustain long glides with passive, low-friction rolling contacts. This structure introduces challenges including nonholonomic rolling constraints, limited traction authority, and coordination of internal forces during push–glide transitions. Bjelonic et al. demonstrated quadrupedal skating on ANYmal using passive wheel end-effectors and force control to generate glide-and-push sequences, reporting reduced cost of transport relative to trotting [19]. SlidBot extended the idea with passive wheels at the knees and lower legs, achieving roller skating gaits with improved speed and energy efficiency on flat and sloped terrain [20]. Chen et al. analyzed passive-wheel quadrupeds and highlighted difficulties in gait transitions and internal force coordination for walking-to-skating switches [21]. Related work also considers interaction with external skateboards [22] and bipedal skating with inline skates [23]. Overall, existing results establish the feasibility of robot skating, but leave open questions in maneuverability, robustness, and systematic design–control integration.

C. Robot Hardware and Control Policy Co-Design

Co-design of morphology and control is a powerful paradigm for unlocking new robot capabilities. Prior works can be broadly grouped into three strategies.

Universal policy methods decouple design and control by training a single design-conditioned policy across a distribution of morphologies, which can then serve as a fast evaluator. Examples include morphology randomization for general locomotion [24], [25] and design-conditioned policies for actuator optimization [26]. These approaches can be versatile, but often require substantial initial training to cover many suboptimal designs.

Joint optimization methods embed morphology parameters directly into the RL loop, updating design and control simultaneously. Such approaches can discover novel morphologies [27], [28], but may suffer from instability due to the non-stationarity induced by changing body parameters and limited simulator support for online morphology changes.

Bilevel optimization methods treat co-design as a nested problem: an outer-loop optimizer proposes designs, and an inner-loop learner trains a specialized policy for each candidate. This paradigm has been applied to legged robots and manipulators [29], [30], producing expert policies tailored to individual morphologies. Because repeated RL training is expensive, sample efficiency in the outer loop is crucial; Bayesian Optimization (BO) is well suited for expensive black-box objectives [31], [8].

Our work adopts this bilevel BO-RL framework and applies it to quadrupedal skating with passive wheels, aiming to jointly discover wheel installation parameters and control policies that yield efficient and versatile behaviors.

III. Quadrupedal Skating via Co-Design

Skating with passive wheels requires both mechanical adaptation and tailored control. Neither component alone is sufficient: wheel orientation shapes which skating gaits are physically feasible, while the learned policy determines whether the robot can stably and efficiently exploit a given design. This motivates a co-design formulation in which hardware and control are optimized jointly. In this section, we formalize the BO-RL bilevel co-design problem, describe the roller support design space, and highlight how quadruped leg kinematics constrain the feasible wheel installation angle.

A. Co-Design via Bilevel Optimization

We formulate quadrupedal skating as the bilevel optimization problem in Fig. 2, with an inner loop for control policy learning and an outer loop for hardware design. For a fixed design $\mathbf{d}$ and a task set $\mathcal{T}$, the inner loop seeks an optimal policy $\pi_\theta^*(\mathbf{d}, \mathcal{T})$ (parameterized by θ) that maximizes expected discounted return over trajectories induced by executing tasks sampled from $\mathcal{T}$:

$$\pi_\theta^*(\mathbf{d}, \mathcal{T}) = \arg\max_{\pi_\theta} \mathbb{E}_{\tau \sim p(\tau | \pi_\theta, \mathbf{d}, \mathcal{T})} \left[\sum_{t=0}^{T} \gamma^t r_t \right], \qquad (1)$$

where τ denotes trajectories generated by controlling the robot with design $\mathbf{d}$ using policy π_θ, r_t is the reward at time step t, and γ is the discount factor. We solve the inner-loop problem with RL. Compared with model-based approaches, RL is well suited to skating because the nonholonomic wheel-ground interactions are handled directly by the simulator, and complex gaits need not be manually specified.

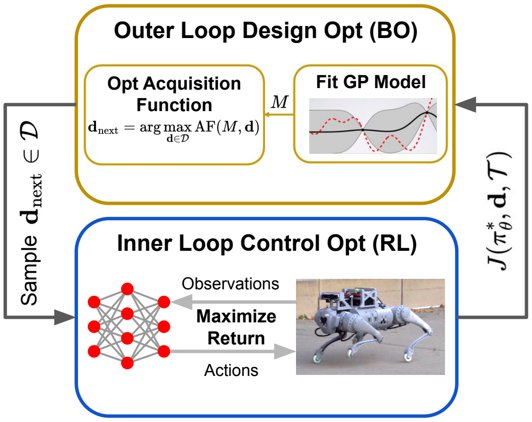
Outer Loop Design Opt (BO)
Opt Acquisition Function
$\mathbf{d}_{\text{next}} = \arg\max_{\mathbf{d}\in\mathcal{D}} \mathrm{AF}(M, \mathbf{d})$
Fit GP Model
M
Sample $\mathbf{d}_{\text{next}} \in \mathcal{D}$
$J(\pi_\theta^*, \mathbf{d}, \mathcal{T})$
Inner Loop Control Opt (RL)
Observations
Maximize Return
Actions

Fig. 2: Bilevel co-design framework for quadrupedal skating. The outer loop uses BO to propose the next candidate design $\mathbf{d}_{\text{next}}$. For each candidate design, the inner loop uses RL to optimize the policy π_θ. The resulting performance $J(\pi_\theta^*, \mathbf{d}, \mathcal{T})$ is fed back to BO, enabling efficient search over design-policy pairs that maximize skating performance.

Given the optimized policy, we evaluate performance using a scalar objective $J(\pi_\theta^*, \mathbf{d}, \mathcal{T})$. The outer loop then searches for the best design within the design space $\mathcal{D}$:

$$\mathbf{d}^* = \arg\min_{\mathbf{d}\in\mathcal{D}} \; J(\pi_\theta^*, \mathbf{d}, \mathcal{T}). \tag{2}$$

Because evaluating $J(\pi_\theta^*, \mathbf{d}, \mathcal{T})$ requires training a full policy and is non-differentiable due to stochastic RL and contact dynamics, the outer loop is naturally treated as black-box optimization. We therefore use BO in the outer loop, which iteratively fits a Gaussain Process (GP) model M to predict the mean and uncertainty of $-J(\pi_\theta^*, \mathbf{d}, \mathcal{T})$, and maximizes an acquisition function $\mathrm{AF}(M, \mathbf{d})$ to select the next candidate $\mathbf{d}_{\text{next}}$ while balancing exploration and exploitation. This bilevel formulation enables systematic discovery of design-policy pairs that yield robust and efficient skating.

B. Mechanical Design of Roller Support

A key component of our platform is a skating foot that mounts a passive wheel at the foot tip of each leg. We replace the robot toes with lightweight 3D-printed supports that connect the legs to the wheels. Our main design parameter is the yaw angle ψ, which determines each wheel's rolling direction (Fig. 1).

A simple but naive design is the *Parallel Configuration* (P configuration) illustrated in Fig. 3, where all wheels are aligned with the body x-axis ($\psi = 0°$). Although this appears ideal for forward skating, the leg kinematics of most state of the art quadrupedal robots–including MIT MiniCheetah [32] and Unitree Go1 [33], Go2 [34], and B2 [35]–constrains the wheel rolling direction to always align with the x-axis when $\psi = 0°$. As a result, the available traction force along the x direction is almost zero due to negligible rolling friction, making v_x effectively uncontrollable. Consequently, at least one wheel must have a nonzero yaw angle, making the optimal design non-trivial and motivating a systematic co-design approach.

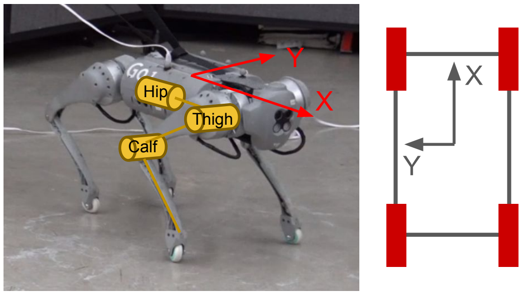

Fig. 3: Naive P configuration ($\psi = 0°$) on Unitree Go1 [33] yields uncontrollable forward velocity v_x due to leg kinematic constraints. Similar limitations arise on many state-of-the-art quadrupedal robots.

IV. Algorithmic Implementation

A. Inner Loop Control Policy Optimization via RL

RL formulation: For each candidate design, we train a control policy with RL in *IsaacLab* [36]. We formulate the learning problem as a Partially Observable Markov Decision Process (POMDP) defined by the tuple $(\mathcal{S}, \mathcal{O}, \mathcal{A}, p, r)$, where $\mathcal{S}$, $\mathcal{O}$, and $\mathcal{A}$ denote the state, observation, and action spaces, respectively, $p(s_{t+1} \mid s_t, a_t)$ is the state transition probability, and r is the reward.

We seek a policy $\pi_\theta(a_t \mid o_t)$ parameterized by θ that maps observations $o_t \in \mathcal{O}$ to a distribution over actions $a_t \in \mathcal{A}$ by maximizing the expected discounted return:

$$\pi_\theta^* = \arg\max_\theta \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right], \tag{3}$$

where $\gamma \in (0, 1)$ is the discount factor. We train π_θ using Proximal Policy Optimization (PPO) [37]. The observation space, action space, and reward terms are summarized below.

Observation Space: Table I lists the observations provided to the policy: commanded and measured base linear/angular velocities, joint positions/velocities, and the previous action. The base orientation is represented by projected gravity ${}^{\mathcal{B}}\mathbf{u}_g = \mathbf{R}_{\mathcal{WB}}[0, 0, -1]^T$, where $\mathbf{R}_{\mathcal{WB}}$ is the rotation of the body frame $\mathcal{B}$ with respect to the world frame $\mathcal{W}$.

A roller-skating-specific detail is how linear velocity commands are specified and, consequently, how they appear in the observation. (4) gives the observation of the commanded linear velocity ${}^{\mathcal{B}}\mathbf{v}_{xy}^{\text{obs}}$ (always expressed in the base frame), which can originate from either a base-frame command ${}^{\mathcal{B}}\mathbf{v}_{xy}^{\text{cmd}}$ or a world-frame command ${}^{\mathcal{W}}\mathbf{v}_{xy}^{\text{cmd}}$ that is transformed into the base frame. These two command conventions lead to different tracking rewards. We refer to the two conventions as *Base Frame Command* and *World Frame Command*:

$${}^{\mathcal{B}}\mathbf{v}_{xy}^{\text{obs}} = \begin{cases} {}^{\mathcal{B}}\mathbf{v}_{xy}^{\text{cmd}} & \text{(Base Frame Command)} \\ \mathbf{R}_{\mathcal{WB}}^T \, {}^{\mathcal{W}}\mathbf{v}_{xy}^{\text{cmd}} & \text{(World Frame Command)} \end{cases}. \tag{4}$$

We compare these two formulations in Sec. V-C.

Action Space: The policy outputs actions $\mathbf{a}$, which are scaled and biased to obtain the desired joint positions $\mathbf{q}^{\text{cmd}}$. These targets are tracked by a low-level joint PD controller with

TABLE I: Observation space.

Name	Expression
Base Velocity	${}^{\mathcal{B}}\mathbf{v}$, ${}^{\mathcal{B}}\boldsymbol{\omega}$
Commanded Base Velocity	${}^{\mathcal{B}}\mathbf{v}^d_{xy}$, ${}^{\mathcal{B}}\omega^d_z$
Projected Gravity	${}^{\mathcal{B}}\mathbf{u}_g$
Joint Position and Velocity	$\mathbf{q}, \dot{\mathbf{q}}$
Last Actions	$\mathbf{a}_{\text{prev}}$

TABLE II: Reward terms.

Name	Expression
Base Linear Velocity xy	$r_{\mathbf{v}_{xy}}$ using (6a) or (7a)
Base Angular Velocity z	r_{ω_z} using (6b) or (7c)
Base Height	$(z - z^d)^2$
Base Orientation	$\left\|{}^{\mathcal{B}}\mathbf{u}_{g,xy}\right\|_2^2$
Base Linear Velocity z	${}^{\mathcal{B}}v_z^2$
Base Angular Velocity xy	$\left\|{}^{\mathcal{B}}\boldsymbol{\omega}_{xy}\right\|_2^2$
Action Rate Minimization	$\|\mathbf{a} - \mathbf{a}_{\text{prev}}\|_2^2$
Joint Acceleration Minimization	$\|\ddot{\mathbf{q}}\|_2^2$
Vertical Virtual Leg	$\left\|\mathbf{p}_{\text{virt leg, xy}}\right\|_2^2$
Collision Avoidance	$\sum \mathbf{1}_{\{\|\mathbf{f}\|_2 > f_0\}}$
Joint Position Limit	$\sum \|\mathbf{q}_{\text{exceed}}\|_2^2$

gains k_p and k_d (and zero desired joint velocity), causing the joint torques $\boldsymbol{\tau}$ to be

$$\boldsymbol{\tau} = k_p(\mathbf{q}^{\text{cmd}} - \mathbf{q}) - k_d\dot{\mathbf{q}}. \tag{5}$$

Rewards: The reward comprises (i) tracking terms for the base velocity, height, and orientation, and (ii) regularization terms that smooth actions and joint accelerations. To discourage overstretched legs, we penalize the horizontal component of the virtual-leg vector (the vector $\mathbf{p}_{\text{virt leg}}$ from hip to wheel) via $\|\mathbf{p}_{\text{virt leg, xy}}\|_2^2$. We also penalize large collision forces $\mathbf{f}$ using $\sum \mathbf{1}_{\{\|\mathbf{f}\|_2 > f_0\}}$ where $\mathbf{1}_{\{\cdot\}}$ is one when $\{\cdot\}$ is true and zero otherwise. We also discourage joint position violations $\mathbf{q}_{\text{exceed}}$ by penalizing $\sum \|\mathbf{q}_{\text{exceed}}\|_2^2$.

Most reward terms are shared across experiments, but the velocity-tracking terms depend on whether commands are specified in the base frame or world frame. For *Base Frame Command*, we use the standard base frame tracking rewards

$$r_{\mathbf{v}_{xy}} = r_{\exp}\left(\left\|{}^{\mathcal{B}}\mathbf{v}_{xy} - {}^{\mathcal{B}}\mathbf{v}^{\text{cmd}}_{xy}\right\|_2\right), \tag{6a}$$

$$r_{\omega_z} = r_{\exp}({}^{\mathcal{B}}\omega_z - {}^{\mathcal{B}}\omega_z^{\text{cmd}}), \tag{6b}$$

where $r_{\exp}(e) = \exp(-e^2/\sigma)$ denotes an exponential tracking reward for the scalar error e.

For *World Frame Command*, we track linear velocity in the world frame and down-weight yaw-rate tracking when the linear velocity tracking error is large:

$$r_{\mathbf{v}_{xy}} = r_{\exp}\left(\left\|{}^{\mathcal{W}}\mathbf{v}_{xy} - {}^{\mathcal{W}}\mathbf{v}^{\text{cmd}}_{xy}\right\|_2\right), \tag{7a}$$

$$r_{\omega_z} = k\, r_{\exp}({}^{\mathcal{B}}\omega_z - {}^{\mathcal{B}}\omega_z^{\text{cmd}}), \tag{7b}$$

$$k = \begin{cases} 1 & ({}^{\mathcal{W}}v^{\text{err}}_{xy} \le e_0) \\ \frac{e_1 - v_{xy,\text{err}}}{e_1 - e_0} & (e_0 < {}^{\mathcal{W}}v^{\text{err}}_{xy} \le e_1) \\ 0 & ({}^{\mathcal{W}}v^{\text{err}}_{xy} > e_1) \end{cases}, \tag{7c}$$

where ${}^{\mathcal{W}}v^{\text{err}}_{xy} := \left\|{}^{\mathcal{W}}\mathbf{v}_{xy} - {}^{\mathcal{W}}\mathbf{v}^{\text{cmd}}_{xy}\right\|_2$ is the world-frame linear velocity tracking error and e_0, e_1 are small/large error thresholds. Fig. 4 illustrates this prioritization. It enables strategic body turning, which is detailed in Sec. V-C.

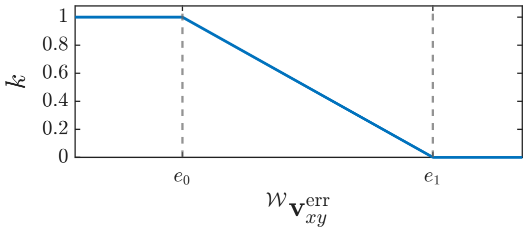

Fig. 4: Illustration of *World Frame Command* angular velocity tracking reward scaling to prioritize linear velocity tracking.

B. Outer Loop Design Optimization via BO

BO Formulation: Given metric evaluations $J(\pi_\theta^*, \mathbf{d}, \mathcal{T})$ from previously tested designs, each outer-loop BO iteration performs two steps: (i) fit a Gaussian Process (GP) surrogate model M to predict the mean and uncertainty of $-J(\pi_\theta^*, \mathbf{d}, \mathcal{T})$, and (ii) maximize an acquisition function AF$(M, \mathbf{d})$ to select the next candidate $\mathbf{d}_{\text{next}}$ while balancing exploration and exploitation. We use a phased acquisition schedule: we begin with Upper Confidence Bound (UCB) and a high exploration coefficient to cover the search space, anneal the exploration coefficient to focus on promising regions, and finally switch to Expected Improvement (EI) to refine the best candidates and improve convergence.

Design Space: The most general parameterization assigns an independent wheel yaw angle to each leg, $\mathbf{d} = [\psi_{FR}, \psi_{FL}, \psi_{RR}, \psi_{RL}]$, where FR, FL, RR, and RL denote front-right, front-left, rear-right, and rear-left, respectively. Exploiting the left–right symmetry yields $\psi_{FR} = -\psi_{FL}$ and $\psi_{RR} = -\psi_{RL}$, reducing the search to a 2D space parameterized by $\psi_{\text{front}} := \psi_{FL}$ and $\psi_{\text{rear}} := \psi_{RL}$. If the front–rear asymmetry is negligible, we further couple all legs as $[\psi_{FR}, \psi_{FL}, \psi_{RR}, \psi_{RL}] = [-\psi, \psi, \psi, -\psi]$, as illustrated in Fig. 5. We report 1D results in Sec. V-A–Sec. V-B and 2D results in Sec. V-D.

Design Metric: To assess a design $\mathbf{d}$ over commands sampled from $\mathcal{T}$ and to average the stochasticity of the RL, we define an instantaneous metric $f(s, a, c)$ as a function of state s, action a and command c in a single time step. Using the rollouts generated during PPO training in IsaacLab, we estimate the design objective $J(\pi_\theta^*, \mathbf{d}, \mathcal{T})$ by Monte Carlo averaging:

$$J(\pi_\theta^*, \mathbf{d}, \mathcal{T}) = \frac{1}{N_{\text{step}} N_{\text{env}}} \sum_{k=1}^{N_{\text{step}}} \sum_{i=1}^{N_{\text{env}}} f(s_{i,k}, a_{i,k}, c_{i,k}), \tag{8}$$

which averages over N_{env} parallel environments and N_{step} time steps. Commands are resampled from $\mathcal{T}$ multiple times within the N_{step}-step window. The simulator time step is 0.02 s.

In practice, we aggregate metrics over windows of $N_{\text{step}} = 1000$ steps ($\approx$ 20 s) across $N_{\text{env}} = 4096$ parallel environments. Velocity commands are resampled every 200 steps, and environments are reset asynchronously. Although PPO updates occur every 24 steps ($\approx$ 0.48 s), averaging metrics over 1000-step windows yields more stable estimates.

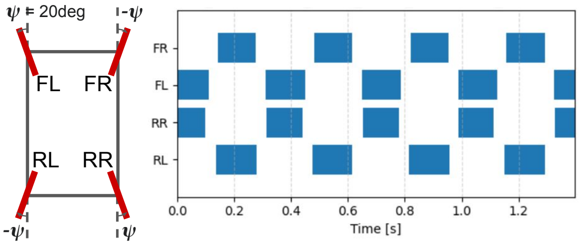

Fig. 5: Illustration of the human-engineered roller wheel angles and the resultant gait pattern.

We instantiate f as an energy-efficiency measure based on the cost of transport (CoT). CoT is computed as the squared joint torque (proportional to motor electrical power) normalized by gravitational force and the ℓ_2 norm of the actual planar twist $\boldsymbol{\xi} = [v_x, v_y, \omega_z]$:

$$\text{CoT} = \frac{\|\boldsymbol{\tau}\|_2^2}{mg\ \|\boldsymbol{\xi}\|_2}, \tag{9}$$

where $\boldsymbol{\tau}$ is the joint-torque vector, m is the mass of the robot, and g is the gravitational acceleration. In the Body Frame Command case, $\boldsymbol{\xi} = [{}^{\mathcal{B}}v_x, {}^{\mathcal{B}}v_y, {}^{\mathcal{B}}\omega_z]$. In the *World Frame Command* case, $\boldsymbol{\xi} = [{}^{\mathcal{W}}v_x, {}^{\mathcal{W}}v_y, {}^{\mathcal{B}}\omega_z]$.

V. Results and Analysis

We present four experiments that evaluate our bilevel BO-RL framework for quadrupedal roller skating. In Sec. V-A, we establish a baseline using a human-engineered wheel design and a control policy trained with *Base Frame Command* tracking. In Sec. V-B, we show that hardware-control co-design is necessary to achieve energy-efficient skating. In Sec. V-C, we demonstrate transient skating behaviors enabled by a *World Frame Command* tracking reward. Finally, in Sec. V-D, we combine co-design (Sec. V-B) and reward engineering (Sec. V-C) to achieve low energy consumption and strong transient velocity tracking.

A. Baseline via Human-Engineered Roller Skating

This section describes the wheel yaw angles of the human-engineered design as the baseline. To address the controllability issue discussed in Sec. III-B, we introduce a minimal design modification in which all wheel angles are parameterized by a single non-zero yaw angle ψ. As shown in Fig. 5, the resulting gait is a trotting gait, where the diagonal legs alternate to contact with the ground. Intuitively, the robot couples each diagonal leg pair to act like a single roller skate.

B. Necessity of Co-Design for Energy Efficiency

We next demonstrate the importance of co-optimizing design and control. In the inner-loop RL used to train control policies, we use *Base Frame Command* observations and rewards, i.e., we specify the velocity command relative to the robot's orientation. This choice enables consistent comparisons of energy efficiency in specific command directions. In the outer-loop BO, we optimize over the minimal 1D design space defined by the yaw angle ψ of the wheels, which generates the four yaw angles of the wheels shown in Fig. 5. The BO objective is the average CoT defined in (8)–(9).

Fig. 8 reports steady-state CoT in polar coordinates while tracking a body-frame linear velocity command with magnitude 1.5 m/s in different directions. The polar angle is the command direction $\alpha = \arctan 2({}^{\mathcal{B}}v_x^{\text{cmd}}, {}^{\mathcal{B}}v_y^{\text{cmd}})$, and the radius is the corresponding CoT. The left subfigure shows that the 1D BO-RL co-designed solution is more energy efficient than walking in almost all directions, whereas the human-engineered design is only more efficient in the forward direction ($\alpha = 0\,\text{deg}$). This indicates that co-optimizing design and control is critical for synthesizing energy-efficient quadrupedal roller skating behaviors.

C. Comparison between Base Frame Command and World Frame Command

We now study the effect of tracking linear velocity in the world frame rather than the base frame. With wheel installation angles fixed to the human-engineered design in Fig. 5, the robot's forward direction is close to the wheel rolling directions. This is energy efficient, but it provides limited friction authority to change the robot's velocity. In contrast, the sideways direction offers substantial friction for velocity tracking but is less energy efficient because the wheels can barely roll laterally.

World Frame Command allows the policy to strategically switch between these regimes to balance energy efficiency and velocity tracking. Fig. 6 illustrates a scenario in which the robot moves quickly and is suddenly commanded to stop. During steady-state motion, the policy aligns the robot's forward direction with the velocity direction to maximize efficiency. When commanded to stop, the robot rotates to align its sideways direction with the velocity direction, producing a large lateral friction force for deceleration. The resulting behavior resembles a pivoting maneuver: the front-right wheel acts as a pivot, while the front-left and rear-right wheels continue rolling, effectively rotating the body about the pivot wheel. The robot then exploits sliding friction from all wheels to achieve a rapid stop.

This maneuver is known in roller skating as the "hockey stop" and is among the quickest ways to stop. It cannot be learned with *Base Frame Command* because base-frame velocity tracking implicitly constrains the body's orientation relative to the commanded velocity, preventing the policy from reorienting to exploit lateral friction for braking. With *World Frame Command*, the stop time from an initial forward speed of 2 m/s is reduced by approximately 50% compared with the *Base Frame Command* policy. This result highlights how world-frame tracking can induce emergent strategies that select body orientation to maximize control authority.

D. Combining Reward Engineering with Co-Design

Combining *World Frame Command* with the 2D design parameterization in Sec. IV-B yields the design shown in Fig. 9. This design exhibits a self-aligning behavior (Fig. 7): without explicitly tracking an angular velocity target, the

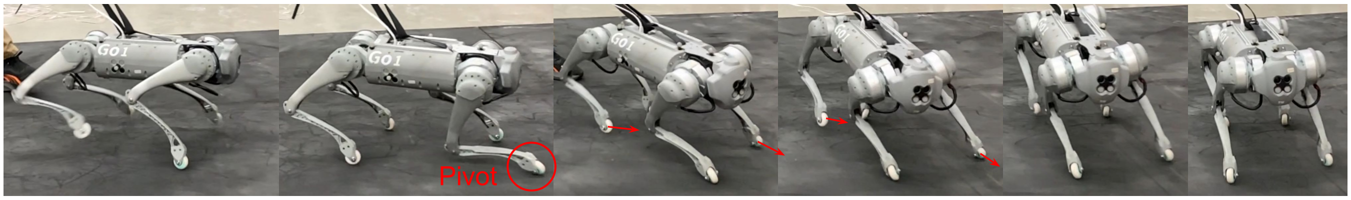

Fig. 6: A rapid stopping strategy known as the “hockey stop” emerged when using *World Frame Command* with the human-engineered design in Fig. 5. When the robot is moving forward quickly and is suddenly commanded to stop, it rotates its body so that its lateral direction (which corresponds to the direction of maximal wheel friction) aligns with the velocity direction, thereby generating a large lateral friction force for deceleration.

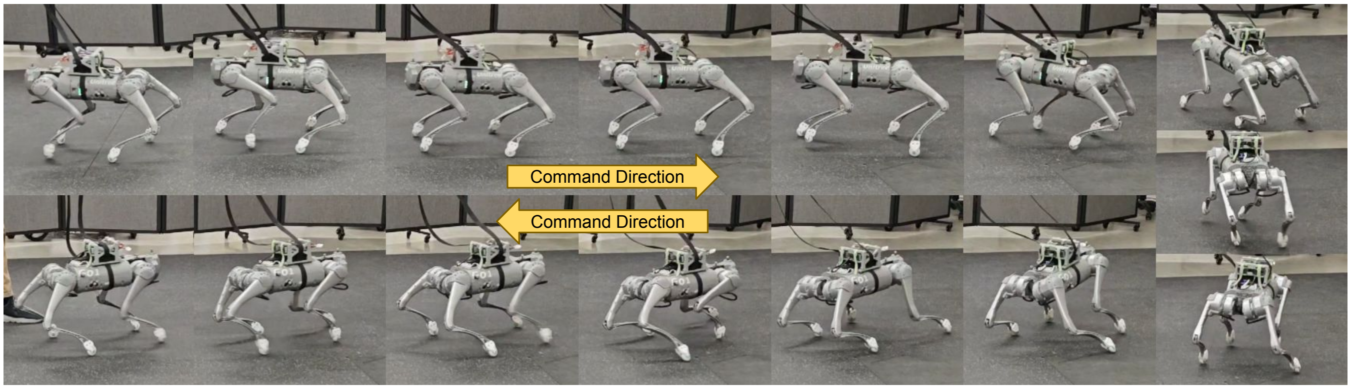

Fig. 7: An energy-efficient self-aligning behavior emerged when combining *World Frame Command* with the 2D design parameterization. For the optimal design in Fig. 9, the backward direction is the most energy-efficient. As a result, the robot learns to consistently align its backward direction with the commanded velocity direction.

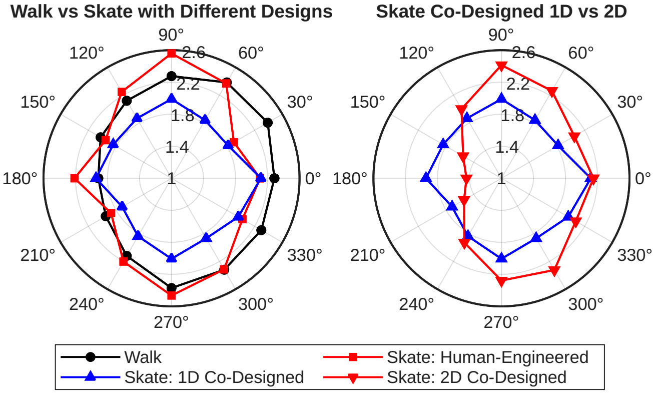

Fig. 8: Directional CoT comparison. The left figure compares the CoT of walking and skating using human-engineered and 1D co-designed roller wheels. The right figure compares skating with 1D and 2D co-designed roller wheels.

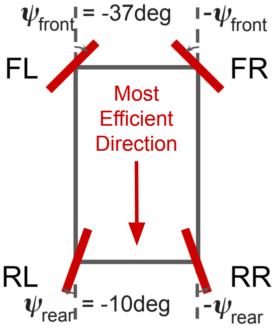

Fig. 9: 2D co-design optimal design.

robot learns to align its backward direction with the commanded linear velocity. This reduces average CoT by 14.6% relative to the 1D co-design.

To understand this mechanism, we fix the 2D-optimized design and retrain the control policy with *Base Frame Command* to evaluate directional CoT. As shown in the right subfigure of Fig. 8, the 2D co-optimized design is less efficient for sideways skating but achieves its lowest CoT in the backward direction—the orientation that the *World Frame Command* policy preferentially aligns with during execution. These results suggest that, when reward design encourages strategic body-orientation selection, the morphology can be optimized for peak efficiency in a preferred direction rather than uniform efficiency across directions, yielding the most energy-efficient skating behavior in this work.

VI. Conclusion and Future Work

We presented quadrupedal skating with passive wheels as a new locomotion mode that merges the efficiency of wheeled motion with the agility of legged robots. Through a bilevel BO-RL co-design framework, we identified design–policy pairs that enable efficient skating and emergent behaviors such as hockey stop and self-aligning motion.

Future work will extend this study beyond flat terrain to outdoor and uneven environments, explore lightweight and durable wheel modules, synthesize better reward engineering, and develop more sample-efficient co-design methods. Integrating skating with walking and running modes is another exciting direction toward versatile and multimodal robots.

References

[1] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, “Bigdog, the rough-terrain quadruped robot,” *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 10 822–10 825, 2008.

[2] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, “Learning agile and dynamic motor skills for legged robots,” *Science robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[3] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, “Dynamic locomotion in the mit cheetah 3 through convex model-predictive control,” in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 1–9.

[4] S. Kim, J. E. Clark, and M. R. Cutkosky, “isprawl: Design and tuning for high-speed autonomous open-loop running,” *The International Journal of Robotics Research*, vol. 25, no. 9, pp. 903–912, 2006.

[5] D. Wettergreen, C. Thorpe, and R. Whittaker, “Exploring mount erebus by walking robot,” *Robotics and Autonomous Systems*, vol. 11, no. 3-4, pp. 171–185, 1993.

[6] C. D. Bellicoso, F. Jenelten, C. Gehring, and M. Hutter, “Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2261–2268, 2018.

[7] M. Bjelonic, C. D. Bellicoso, Y. de Viragh, D. Sako, F. D. Tresoldi, F. Jenelten, and M. Hutter, “Keep rollin’—whole-body motion control and planning for wheeled quadrupedal robots,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2116–2123, 2019.

[8] E. Brochu, V. M. Cora, and N. De Freitas, “A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning,” *arXiv preprint arXiv:1012.2599*, 2010.

[9] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, “Sim-to-real: Learning agile locomotion for quadruped robots,” *arXiv preprint arXiv:1804.10332*, 2018.

[10] X. B. Peng, G. Berseth, K. Yin, and M. Van De Panne, “Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning,” *Acm transactions on graphics (tog)*, vol. 36, no. 4, pp. 1–13, 2017.

[11] S. Seok, A. Wang, M. Y. Chuah, D. Otten, J. Lang, and S. Kim, “Design principles for highly efficient quadrupeds and implementation on the mit cheetah robot,” in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3307–3312.

[12] L. Bao, J. Humphreys, T. Peng, and C. Zhou, “Deep reinforcement learning for bipedal locomotion: a brief survey (2024),” *arXiv preprint arXiv:2404.17070*.

[13] S. Ha, J. Lee, M. van de Panne, Z. Xie, W. Yu, and M. Khadiv, “Learning-based legged locomotion: State of the art and future perspectives,” *The International Journal of Robotics Research*, vol. 44, no. 8, pp. 1396–1427, 2025.

[14] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch *et al.*, “Anymal-a highly mobile and dynamic quadrupedal robot,” in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2016, pp. 38–44.

[15] L. Cui, S. Wang, J. Zhang, D. Zhang, J. Lai, Y. Zheng, Z. Zhang, and Z.-P. Jiang, “Learning-based balance control of wheel-legged robots,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7667–7674, 2021.

[16] R. Li, H. Wang, Q. Li, Z. Han, Y. Chu, L. Ye, W. Xie, and W. Liao, “Ctbc: Contact-triggered blind climbing for wheeled bipedal robots with instruction learning and reinforcement learning,” *arXiv preprint arXiv:2509.02986*, 2025.

[17] K. G. Gim and J. Kim, “Ringbot: Monocycle robot with legs,” *IEEE Transactions on Robotics*, vol. 40, pp. 1890–1905, 2024.

[18] M. Bjelonic, V. Klemm, J. Lee, and M. Hutter, “A survey of wheeled-legged robots,” in *Climbing and walking robots conference*. Springer, 2022, pp. 83–94.

[19] M. Bjelonic, C. D. Bellicoso, M. E. Tiryaki, and M. Hutter, “Skating with a force controlled quadrupedal robot,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7555–7561.

[20] Z. Wei, J. Ren, J. Guo, Y. Yang, S. Xiang, D. Chen, J. Liu, and A. Song, “Slidbot: A quadruped robot with passive wheels for roller skating,” *Journal of Bionic Engineering*, vol. 22, no. 6, pp. 2831–2848, 2025.

[21] J. Chen, K. Xu, and X. Ding, “Roller-skating of mammalian quadrupedal robot with passive wheels inspired by human,” *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 3, pp. 1624–1634, 2020.

[22] H. Liu, S. Teng, B. Liu, W. Zhang, and M. Ghaffari, “Discrete-time hybrid automata learning: Legged locomotion meets skateboarding,” *arXiv preprint arXiv:2503.01842*, 2025.

[23] N. Ziv, Y. K. Lee, and G. Ciaravella, “Motion generation for a humanoid robot with inline-skate.” in *ICINCO (2)*, 2010, pp. 354–359.

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

[25] J. Won and J. Lee, “Learning body shape variation in physics-based characters,” *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–12, 2019.

[26] F. Bjelonic, J. Lee, P. Arm, D. Sako, D. Tateo, J. Peters, and M. Hutter, “Learning-based design and control for quadrupedal robots with parallel-elastic actuators,” *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1611–1618, 2023.

[27] C. Schaff, D. Yunis, A. Chakrabarti, and M. R. Walter, “Jointly learning to construct and control agents using deep reinforcement learning,” in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 9798–9805.

[28] D. Ha, “Reinforcement learning for improving agent design,” *Artificial life*, vol. 25, no. 4, pp. 352–365, 2019.

[29] C. Chen, P. Xiang, J. Zhang, R. Xiong, Y. Wang, and H. Lu, “Deep reinforcement learning based co-optimization of morphology and gait for small-scale legged robot,” *IEEE/ASME Transactions on Mechatronics*, vol. 29, no. 4, pp. 2697–2708, 2023.

[30] Y. Kim, Z. Pan, and K. Hauser, “Mo-bbo: Multi-objective bilevel bayesian optimization for robot and behavior co-design,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9877–9883.

[31] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, “Taking the human out of the loop: A review of bayesian optimization,” *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[32] B. Katz, J. Di Carlo, and S. Kim, “Mini cheetah: A platform for pushing the limits of dynamic quadruped control,” in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6295–6301.

[33] Unitree Go1: `https://www.unitree.com/go1`.

[34] Unitree Go2: `https://www.unitree.com/go2`.

[35] Unitree B2: `https://www.unitree.com/b2`.

[36] M. Mittal, P. Roth, J. Tigue, A. Richard, O. Zhang, P. Du, A. Serrano-Munoz, X. Yao, R. Zurbrügg, N. Rudin *et al.*, “Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning,” *arXiv preprint arXiv:2511.04831*, 2025.

[37] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.