2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Phase-Aware Policy Learning for Skateboard Riding of Quadruped Robots via Feature-wise Linear Modulation

Minsung Yoon*, Jeil Jeong*, Sung-Eui Yoon†

***Abstract*— Skateboards offer a compact and efficient means of transportation as a type of personal mobility device. However, controlling them with legged robots poses several challenges for policy learning due to perception-driven interactions and multi-modal control objectives across distinct skateboarding phases. To address these challenges, we introduce Phase-Aware Policy Learning (*PAPL*), a reinforcement-learning framework tailored for skateboarding with quadruped robots. *PAPL* leverages the cyclic nature of skateboarding by integrating phase-conditioned Feature-wise Linear Modulation layers into actor and critic networks, enabling a unified policy that captures phase-dependent behaviors while sharing robot-specific knowledge across phases. Our evaluations in simulation validate command-tracking accuracy and conduct ablation studies quantifying each component's contribution. We also compare locomotion efficiency against leg and wheel–leg baselines and show the real-world transferability.**

I. Introduction

Legged robots have shown robust locomotion across challenging terrains, including icy, rough, and deformable surfaces [1]–[3]. However, their leg-based actuation inherently limits speed and energy efficiency, especially for long-range missions under constrained battery capacity [4]. To mitigate this, recent studies have investigated augmenting robots with riding capabilities [5]–[12]. This modality allows the robot to leverage personal mobility devices, such as skateboards or Segways, enabling more efficient long-distance travel while conserving onboard energy, analogous to human behavior.

Among these devices, the skateboard is a lightweight, non-motorized platform that offers distinct advantages for legged robots without manipulators. Propulsion arises from forces applied to the deck—whether from the rider's kicking impact or gravity—which are converted into wheel torques through wheel–ground interaction, while the low rolling resistance of the bearings minimizes energy loss. Furthermore, the skateboard's truck assembly allows steering via weight shifting on the deck, eliminating the need for grasp-based manipulation. Sec. III provides a detailed description of these mechanisms.

Despite these advantages, the design of skateboard-riding controllers for legged robots remains challenging. The skateboarding motion is multi-modal, comprising cyclic pushing and carving phases, and transitions where the feet designated for propulsion alternate between the ground and the board. Each phase entails distinct contact patterns, dynamics, and control objectives, with seamless transitions further complicating the design of a unified controller. In addition, the robot must sustain balance against non-inertial forces induced by the board's acceleration and avoid desynchronization caused by prolonged ground contact. Moreover, persistent stability requires exteroceptive–control coupling for synchronization and recovery from unstable states through posture adaptation.

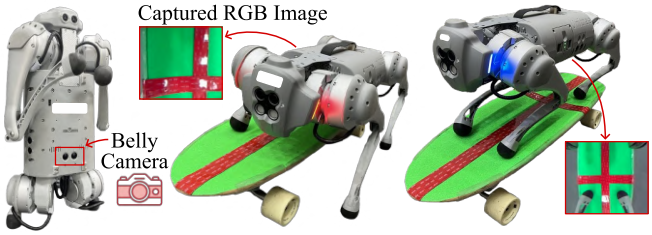

Fig. 1. Belly-mounted RGB camera setup on the Unitree Go1 robot [13]. The camera observes the skateboard deck surface and supplies visual feedback for localization and control, enabling resilient skateboarding maneuvers.

Recent studies have applied model-based optimization [11] and Reinforcement Learning (RL) [12] for skateboarding of quadruped robots. The model-based method relies on a pre-computed motion library and a linearized model for tracking, but limited model fidelity and library coverage could reduce robustness to unforeseen events such as slippage. In contrast, the prior RL-based research develops policies directly from interaction, but it simplifies the training setup by assuming a foot–board attachment in simulation, reducing the underactuated dynamics of two floating bases to a coupled system. While this design choice facilitates training, it could reduce generalization at deployment and restrict maneuvers mainly to gliding rather than steering through deck tilting. In this work, we aim to relax such assumptions and integrate exteroceptive perception to enable sustained skateboarding with visual feedback, as illustrated in Fig. 1. To the best of our knowledge, this represents one of the first efforts to utilize onboard exteroceptive sensing for quadruped skateboarding.

We propose Phase-Aware Policy Learning (*PAPL*), an RL framework that leverages the cyclic and multi-modal nature of skateboard riding. *PAPL* incorporates a phase-conditioned modulation mechanism within actor–critic networks, effectively encoding phase-specific behaviors in a unified policy. To induce proficient riding skills under partial observability, we employ a two-stage scheme of asymmetric privileged learning and distillation, improving sample efficiency by exploiting privileged states in simulation and later replacing them with estimates from observation history. Estimating the robot-relative skateboard state enables real-time posture adaptation for resilient skateboarding. In experiments, we assess command-tracking accuracy, analyze *PAPL* components via ablation studies, compare locomotion efficiency against legged and wheel–legged baselines, and validate sim-to-real transferability across diverse environmental conditions.

*These authors contributed equally to this work. All authors are with the School of Computing at the Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea. †S. Yoon is a corresponding author (e-mail: sungeui@kaist.edu).

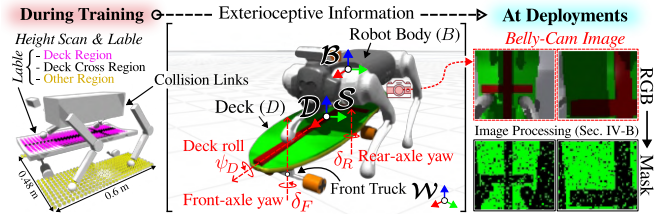

Fig. 2. The center shows the reference frames: robot body $\mathcal{B}$, skateboard $\mathcal{S}$, deck $\mathcal{D}$, and world $\mathcal{W}$, along with the physical robot body B and deck D objects. The skateboard frame $\mathcal{S}$ is fixed to the board, with the deck $\mathcal{D}$ rotating about its roll axis relative to $\mathcal{S}$; joint variables include deck roll (ψ_D) and the yaw angles of the front and rear wheel axles (δ_F, δ_R). The left and right show exteroceptive inputs for training and deployment stages.

II. Variable Notation

In Cartesian space, translational variables such as position $\mathbf{p}$, linear velocity $\mathbf{v}$, acceleration $\dot{\mathbf{v}}$, and force $\mathbf{f}$, as well as rotational variables including XYZ Euler angles $\boldsymbol{\theta}$, angular velocity $\boldsymbol{\omega}$, angular acceleration $\dot{\boldsymbol{\omega}}$, and torque $\boldsymbol{\tau}$, are defined in $\mathbb{R}^3$. For clarity, we employ superscripts to denote reference frames and subscripts to specify the object, coordinate, and time index. For instance, $p^{\mathcal{S}}_{B,x,t-1}$ denotes the x-component of the body's position p_B represented in the skateboard frame $\mathcal{S}$ at time $t-1$. For brevity, we omit the current time index t. For quadruped robots with 12 actuated joints, joint position $\boldsymbol{q}$, velocity $\dot{\boldsymbol{q}}$, acceleration $\ddot{\boldsymbol{q}}$, and torque $\boldsymbol{\tau}_q$ lie in $\mathbb{R}^{12}$. Each foot F_i, where $i \in \{0,1,2,3\}$, is associated with a contact force $\mathbf{f}_{F_i}$ and binary contact state $c_i \in \{0,1\}$. We define $\mathcal{F}_{\text{all}}$, $\mathcal{F}_{\text{left}}$, and $\mathcal{F}_{\text{right}}$ as the index sets of all feet, left-side feet, and right-side feet, respectively, and $\mathcal{J}_{\text{all}}$, $\mathcal{J}_{\text{left}}$, and $\mathcal{J}_{\text{right}}$ as the corresponding index sets of actuated joints, grouped by leg. For skateboards, representative joints include the deck roll angle ψ_D and yaw angles of the front and rear wheel axes, δ_F and δ_R, all in $\mathbb{R}$, as illustrated in Fig. 2.

III. Skateboard Dynamics Modeling

Accurate modeling of skateboard dynamics is essential to develop riding policies and narrow the sim-to-real gap. Thus, our model captures two mechanisms: steering, resulting from the truck's trigonometric geometry; and propulsion, which is inherently passive but modeled by applying wheel torques to compensate for contact modeling limitations in simulation.

A. Steering Dynamics

The skateboard is a passive, non-holonomic system with a rigid deck mounted on front and rear trucks, each connecting a pair of wheels via an axle. Steering arises from a roll-to-yaw coupling: rider-induced deck roll is converted into yaw rotation of the wheel axles. This coupling defines a nonlinear relationship between the deck roll angle ψ_D and the yaw angles of the front and rear axles, δ_F and δ_R, expressed as:

$$\delta_i = \arctan\left(\gamma_i^1 \sin\left(\gamma_i^2 \psi_D\right)\right),\ i \in \{F, R\} \tag{1}$$

where γ_i^1 and γ_i^2 are skateboard parameters determined by the mechanical configuration [14]. We emulate this passive steering mechanism using Proportional-Derivative (PD) controllers at each axle to track the desired angles δ_F and δ_R.

Thus, the rider achieves effective steering by actively modulating the deck roll ψ_D through contact force distribution on the deck. Its roll dynamics, governed by bushing compliance and rider-induced torques, is modeled as follows:

$$\boldsymbol{\tau}_D^{\mathcal{D}} = \sum_{i \in \mathcal{F}_{\text{is-on-deck}}} \mathbf{p}_{F_i}^{\mathcal{D}} \times \mathbf{f}_{F_i}^{\mathcal{D}},$$

$$\dot{\omega}_{D,\text{x}}^{\mathcal{D}} = (-k^{\text{P}}_{\text{bushing}} \psi_D - k^{\text{D}}_{\text{bushing}} \dot{\psi}^D + \tau_{D,\text{x}}^{\mathcal{D}})/I_{D,\text{xx}},$$

where $k^{\text{P}}_{\text{bushing}}$ and $k^{\text{D}}_{\text{bushing}}$ represent the bushing stiffness and damping coefficients of the restoring torque, and $I_{D,\text{xx}}$ is the deck's moment of inertia about its roll axis.

B. Propulsion Dynamics

Physics engines often exhibit limited accuracy in modeling wheel–ground contact and rolling resistance [12]. This inaccuracy prevents external forces applied to the deck from being fully transmitted to the wheels as effective driving torque, resulting in unrealistic propulsion behavior. To mitigate this issue, we explicitly compute and apply effective torques to the wheels. The net torque applied to left and right wheels $j \in \{L, R\}$ of front and rear axles $i \in \{F, R\}$ is defined as

$$\tau_{i,j}^{\text{net}} = \tau_i^{\text{ext}} - \tau_{i,j}^{\text{fric}}.$$

The external torque τ_i^{ext} is computed by projecting the net force acting on the deck along the rolling direction of each axle. The net force in the skateboard frame $\mathcal{S}$ is modeled as

$$\mathbf{f}_D^{\mathcal{S}} = -\sum_{i \in \mathcal{F}_D} \mathbf{f}_{F_i}^{\mathcal{S}} + m_S \mathbf{g}^{\mathcal{S}},$$

where m_S is the skateboard mass and $\mathbf{g}$ is gravitational acceleration. Due to non-holonomic constraints, only the force aligned with the axle directions contributes to propulsion. The axle direction is represented by the unit vector $\hat{\boldsymbol{d}}_i^{\mathcal{S}} = (\cos\delta_i,\ \sin\delta_i,\ 0)^\top$, with the corresponding projected force $f_i^{\text{drive}} = \boldsymbol{f}_D^{\mathcal{S}} \cdot \hat{\boldsymbol{d}}_i^{\mathcal{S}}$ at each axle i. Assuming symmetric torque distribution across wheels, we compute the resulting external torque as $\tau_i^{\text{ext}} = \frac{r}{2} f_i^{\text{drive}}$, where r is the wheel radius.

The friction torque $\tau_{i,j}^{\text{fric}}$ accounts for static friction as well as dynamic effects—including viscous damping and rolling resistance—as a function of the wheel's angular velocity $\omega_{i,j}$:

$$\tau_{i,j}^{\text{fric}} = \begin{cases} \min\left(|\tau_i^{\text{ext}}|,\ \tau_{\text{static}}\right) \cdot \text{sign}\left(\tau_i^{\text{ext}}\right), & |\omega_{i,j}| < \epsilon \\ c_\omega \omega_{i,j} + r\mu_r m_S ||\mathbf{g}||_2 \cdot \text{sign}(\omega_{i,j}), & |\omega_{i,j}| \geq \epsilon \end{cases},$$

where τ_{static} is the static friction limit, μ_r is the rolling resistance coefficient, and c_ω is the viscous damping constant.

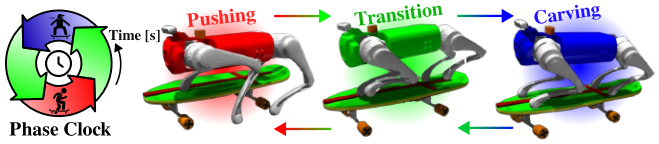

Fig. 3. Illustration of the phase clock concept that manages the cyclic nature of skateboarding, along with representative motion snapshots of each phase.

IV. Skateboard-Riding Policy Learning

We present the Phase-Aware Policy Learning framework (*PAPL*), which enables quadruped robots to ride skateboards. To exploit the riding task's cyclic nature shown in Fig. 3, we integrate a phase clock into the learning process and policy architecture. The following description provides the problem formulation, policy composition, and implementation details.

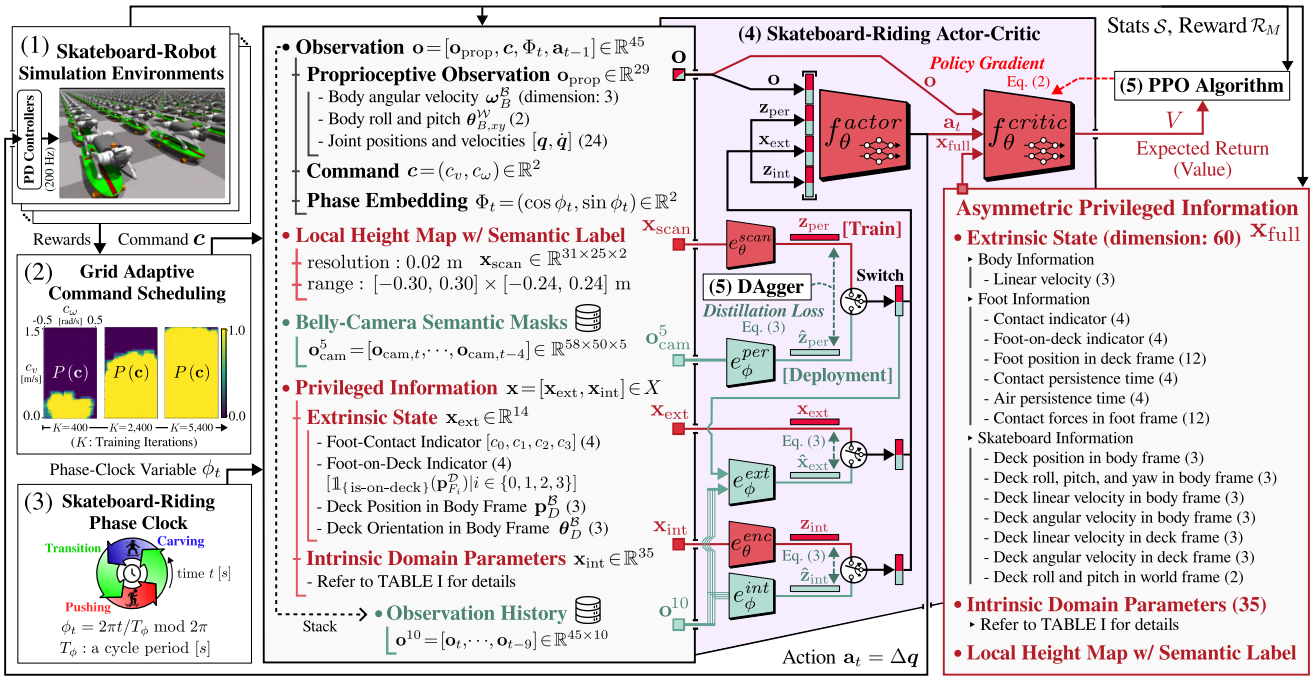

Fig. 4. **Phase-Aware Policy Learning (*PAPL*) Framework for Skateboard Riding.** (1) Simulation environments modeling skateboard–robot interaction. (2) Command scheduling that procedurally increases riding difficulty for broad command-space coverage [15]. (3) Phase-clock representation that alternates over time between pushing, transition, and carving modes. (4) An asymmetric actor–critic architecture: the critic leverages full privileged information for effective policy guidance with clear situational awareness, while the actor relies solely on features that can be inferred or directly observed. (5) Proximal Policy Optimization (PPO) [16] trains policy networks parameterized by θ (red-color networks) to maximize Eq. (2). The converged policy is then distilled via Dataset Aggregation (DAgger) [17] for the estimators parameterized by ϕ (mint) using Eq. (3), replacing inaccessible information during deployment.

A. Formulation of Skateboarding Policy Learning

We formulate a skateboarding task as a phase-conditioned reinforcement learning problem, where the quadruped robot learns to perform riding skills—pushing, mounting, carving, and foot planting—on a dynamic skateboard, as shown in Fig. 3. To represent the riding task's cyclic and multi-modal structure, we introduce a phase clock variable $\phi_t \in [0, 2\pi)$:

$$\phi_t = 2\pi t / T_\phi \bmod 2\pi,$$

where T_ϕ denotes the period of one skateboarding cycle. This cyclic phase variable determines a discrete motion mode M:

$$M(\phi_t) = \begin{cases} \textsc{Carving} & \text{if } \phi_t \in [0.2\pi,\ 0.8\pi], \\ \textsc{Pushing} & \text{if } \phi_t \in [1.2\pi,\ 1.8\pi], \\ \textsc{Transition} & \text{otherwise.} \end{cases}$$

We employ the phase variable ϕ_t to modulate distinct control intents of the policy. Accordingly, we model the task as a phase-conditioned Partially Observable Markov decision process (POMDP), defined by the tuple $(\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{T}, \mathcal{R}_M, \rho_0, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{O} \subset \mathcal{S}$ the observation space, $\mathcal{A}$ the action space, $\mathcal{T}$ the transition dynamics, $\mathcal{R}_M$ the mode-dependent reward function, ρ_0 the initial state distribution, and γ the discount factor. We set the initial state $\boldsymbol{s}_0 \sim \rho_0$ by placing the robot at the center of the skateboard in a nominal crouching posture $\boldsymbol{q}_0$, with small deviations in $\mathbf{p}_B^{\mathcal{S}}$ and $\boldsymbol{q}$. We then optimize the skateboard-riding policy π_θ by maximizing the expected return over commands $\boldsymbol{c}$ and phase periods T_ϕ:

$$J(\theta) = \mathbb{E}_{\substack{\boldsymbol{c}\sim P(\boldsymbol{c}),\\ T_\phi \sim P(T_\phi)}} \left[\mathbb{E}_{\substack{\boldsymbol{s}_0\sim\rho_0,\\ (\boldsymbol{s},\boldsymbol{a})\sim\rho_{\pi_\theta}}} \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}_{M(\phi_t)}(\boldsymbol{s}_t, \boldsymbol{a}_t | \boldsymbol{c}) \right] \right], \quad (2)$$

where ρ_{π_θ} is the state-action visitation distribution under the policy π parameterized by θ, and $\boldsymbol{c} = (c_v, c_\omega) \in \mathbb{R}^2$ denotes a set of forward velocity and yaw rate commands. To facilitate skill acquisition, we employ a grid-adaptive scheduling strategy over the command distribution $P(\boldsymbol{c})$ [15]. As shown in Fig. 4-(2), task difficulty is progressively increased by expanding command space in line with policy proficiency. For the phase-period distribution $P(T_\phi)$, we uniformly sample $T_\phi \sim \mathcal{U}(4.0, 12.0)$ s at the beginning of each episode.

Partial observability in POMDPs introduces challenges in learning complex motor skills through direct policy optimization [18]–[20]. To address this, we adopt a privileged learning framework that reformulates the problem as an MDP using privileged information $X \subset S \setminus O$ [21]–[27]. This privileged state information X provides richer environmental context to the policy, facilitating optimization with respect to Eq. (2).

Specifically, we adopt a two-stage learning procedure that integrates privileged learning with system identification [24]–[26]. In the first stage, we develop a skateboard-riding policy using Proximal Policy Optimization (PPO) [16] along with an asymmetric actor-critic architecture: the critic utilizes the full privileged state available in simulation, while the actor exploits a subset that is inferable from sensor observations. In the second stage, we replace the actor's simulation-only components with estimates to enable policy deployment under partial observability. To this end, we train three complementary estimators using the Dataset Aggregation (DAgger) algorithm [17]. As shown in Fig. 4-(4), the estimators infer inaccessible privileged features from observation history and provide surrogate inputs to the actor network at deployment.

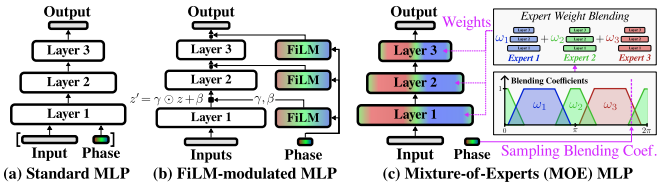

Fig. 5. **Multilayer perceptron (MLP) network variants.** (a) Standard MLP, (b) FiLM-modulated MLP with phase-conditioned feature-wise modulation, and (c) Mixture-of-Experts MLP with phase-based expert weight blending.

B. Phase-Aware Policy Composition

To encode phase-dependent multi-modal behaviors within a unified policy, we compose the actor f_θ^{actor} and critic f_θ^{critic} networks with Feature-wise Linear Modulation (FiLM) [28]–[30], which serves as an effective structural inductive bias for handling strong modality shifts in input–output distributions conditioned on specific variables. As illustrated in Fig. 5-(b), we implement each layer in the FiLM-modulated multilayer perceptron (MLP) $\mathcal{F}$ to apply an affine transformation followed by modulation conditioned on the phase embedding $\Phi_t = (\cos\phi_t, \sin\phi_t)^\top \in \mathbb{R}^2$. The computation at layer ℓ is:

$$z = \mathbf{W}_\ell \mathbf{h}_{\ell-1} + \mathbf{b}_\ell, \quad \mathbf{h}_\ell = \sigma\left(\gamma_\ell(\Phi_t) \odot z + \beta_\ell(\Phi_t)\right)$$

where $\mathbf{h}_\ell$, $\mathbf{W}_\ell$, and $\mathbf{b}_\ell$ denote the layer activation, weight, and bias; $\gamma_\ell(\Phi_t)$ and $\beta_\ell(\Phi_t)$ are phase-conditioned modulation parameters; $\sigma(\cdot)$ is a nonlinear activation function; and $\odot$ is element-wise multiplication. This modulation enables the networks to adapt to phase-specific variations while sharing robot-specific knowledge, facilitating smooth transitions and efficient motor skill reuse across the skateboard-riding cycle.

At each time step, the actor network f_θ^{actor} outputs joint-displacement actions $\Delta\boldsymbol{q} \in \mathcal{A}$, representing deviations from the nominal posture $\boldsymbol{q}_0$. These are converted into torques $\boldsymbol{\tau}_q$ by applying $\boldsymbol{q}_0 + \Delta\boldsymbol{q}$ as targets to the joint PD controllers. The actor takes as input a structured tuple $(\mathbf{o}, \mathbf{z}_{\text{per}}, \mathbf{x}_{\text{ext}}, \mathbf{z}_{\text{int}})$, where the observation $\mathbf{o}$ includes proprioceptive information $\mathbf{o}_{\text{prop}}$, the command $\boldsymbol{c}$, the phase embedding Φ_t, and the previous action $\boldsymbol{a}_{t-1}$; the perceptual feature $\mathbf{z}_{\text{per}}$ is produced by a scan encoder $e_\theta^{scan}: \mathbf{x}_{\text{scan}} \to \mathbf{z}_{\text{per}}$, capturing skateboard-aware local height and semantic information, as depicted in Fig. 2; the extrinsic privileged state $\mathbf{x}_{\text{ext}}$ consists of the deck's pose relative to the robot body as well as foot information; and the intrinsic latent feature $\mathbf{z}_{\text{int}}$ is generated by the encoder $e_\theta^{enc}: \mathbf{x}_{\text{int}} \to \mathbf{z}_{\text{int}}$. Intrinsic parameters $\mathbf{x}_{\text{int}}$ (TABLE I) affect the transition dynamics and contribute to the domain gap.

Term	Dim.	Training Range	Testing Range	Unit
Quadruped Robots				
Payload Mass	(1)	$[0.0, 1.5]$	$[0.0, 3.0]$	kg
Shifted CoM	(3)	$[-0.05, 0.05]$	$[-0.1, 0.1]$	m
Friction Coef.	(1)	$[0.8, 1.2]$	$[0.7, 2.0]$	-
Restitution Coef.	(1)	$[0.0, 0.1]$	$[0.0, 0.15]$	-
Leg-Joint PD Stiffness	(12)	$[36.0, 44.0]$	$[34.0, 46.0]$	N m/rad
Leg-Joint PD Damping	(12)	$[0.8, 1.2]$	$[0.7, 1.3]$	N m s/rad
Skateboards				
Deck Mass	(1)	$[3.5, 4.5]$	$[3.0, 5.0]$	kg
Truck-Yaw PD Stiff.	(1)	$[45.0, 50.0]$	$[40.0, 55.0]$	N m/rad
Truck-Yaw PD Damp.	(1)	$[2.5, 3.0]$	$[2.2, 3.3]$	N m s/rad
Bushing PD Stiffness	(1)	$[30.0, 35.0]$	$[25.0, 40.0]$	N m/rad
Bushing PD Damping	(1)	$[1.8, 2.0]$	$[1.5, 2.3]$	N m s/rad

TABLE I. Domain randomization ranges for intrinsic parameters, $\mathbf{x}_{\text{int}}$.

Net.	Inputs (dimension)	Architecture	Outputs
f_θ^{critic}	$[\mathbf{o}, \mathbf{x}_{\text{scan}}, \mathbf{x}_{\text{full}}]$ (1702)	$\mathcal{F}$(1024, 512, 256, 1)	$[V]$ (1)
f_θ^{actor}	$[\mathbf{o}, \mathbf{z}_{\text{per}}, \mathbf{z}_{\text{int}}, \mathbf{x}_{\text{ext}}]$ (91)	$\mathcal{F}$(512, 256, 128, 12)	$[\mathbf{a}]$ (12)
e_θ^{scan}	$[\mathbf{x}_{\text{scan}}]$ (64 x 86)	1D CNN-GRU + [16]	$[\mathbf{z}_{\text{per}}]$ (16)
e_θ^{enc}	$[\mathbf{x}_{\text{int}}]$ (35)	[128, 64, 16]	$[\mathbf{z}_{\text{int}}]$ (16)
e_ϕ^{per}	$[\mathbf{o}_{\text{cam}}^5]$ (128 x 256 x 5)	2D CNN-GRU + [16]	$[\hat{\mathbf{z}}_{\text{per}}]$ (16)
e_ϕ^{ext}	$[\mathbf{o}^{10}, \hat{\mathbf{z}}_{\text{per}}]$ (45 x 10, 16)	1D CNN-GRU + [14]	$[\hat{\mathbf{x}}_{\text{ext}}]$ (14)
e_ϕ^{int}	$[\mathbf{o}^{10}]$ (45 x 10)	1D CNN-GRU + [16]	$[\hat{\mathbf{z}}_{\text{int}}]$ (16)

TABLE II. **Network Architectures.** $\mathcal{F}(\cdot)$ denotes a FiLM-modulated MLP with intermediate and final layer dimensions. $[\cdot]$ indicates a standard MLP. CNN-GRU represents a convolutional encoder followed by a gated recurrent unit with a 32-dimensional hidden state, encoding spatiotemporal inputs.

For deployment, we replace privileged features $\mathbf{z}_{\text{per}}$, $\mathbf{x}_{\text{ext}}$, $\mathbf{z}_{\text{int}}$ with estimates from their corresponding estimators. The perceptual estimator $e_\phi^{per}: \mathbf{o}_{\text{cam}}^5 \to \hat{\mathbf{z}}_{\text{per}}$ infers perceptual latent features from a sequence of five belly-mounted camera images. To improve robustness to lighting changes and specular reflections, the RGB images are filtered by the deck color in the HSV color space to obtain semantic masks, and randomly perturbed with salt-and-pepper noise or set entirely to zero, as shown in Fig. 2. The extrinsic estimator $e_\phi^{ext}: (\mathbf{o}^{10}, \hat{\mathbf{z}}_{\text{per}}) \to \hat{\mathbf{x}}_{\text{ext}}$ explicitly infers foot states and deck pose relative to the body, while the intrinsic estimator $e_\phi^{int}: \mathbf{o}^{10} \to \hat{\mathbf{z}}_{\text{int}}$ extracts latent domain features from a 10-observation history. These estimators are trained by minimizing the distillation loss:

$$\mathcal{L}_{\text{DAgger}}(\phi) = \mathbb{E}_{\mathcal{D}}\big[\, \|\mathbf{z}_{\text{per}} - \hat{\mathbf{z}}_{\text{per}}(\phi)\|_2^2 + \|\mathbf{x}_{\text{ext}} - \hat{\mathbf{x}}_{\text{ext}}(\phi)\|_2^2 + \|\mathbf{z}_{\text{int}} - \hat{\mathbf{z}}_{\text{int}}(\phi)\|_2^2 \,\big], \quad (3)$$

where ϕ is the estimators' parameters to be optimized. The dataset $\mathcal{D}$ is constructed by iteratively rolling out the policy and annotating the estimators' observations with supervision.

As listed in TABLE III, we define the reward function as $\mathcal{R}_M = \mathcal{R}_{\text{dep}} + \mathcal{R}_{\text{ind}}$, where $\mathcal{R}_{\text{dep}} = \sum_{i=1}^{7} r_i$ contains mode-dependent terms that adapt the policy to each skateboarding mode $M(\phi_t)$. Specifically, r_1–r_4 promote accurate command tracking, body and foot placement, and riding posture, while r_5–r_7 regulate foot slippage, contact patterns, and clearance. The mode-independent component $\mathcal{R}_{\text{ind}} = \sum_{i=8}^{11} r_i$ enforces robot-skateboard alignment (r_8), joint smoothness (r_9), stable movements (r_{10}), and safety (r_{11}) across all phases to ensure energy efficiency, robustness, and safe operation. This structured design enables the phase-conditioned policy to acquire mode-specific skills while maintaining stable behavior throughout the skateboarding cycle. While the TRANSITION mode has no dedicated reward terms, maximizing expected returns induces mounting and dismounting behaviors in $\phi \in [1.8\pi, 2\pi) \cup [0, 0.2\pi]$ and $\phi \in [0.8\pi, 1.2\pi]$, respectively, for the periodically upcoming CARVING and PUSHING modes.

C. Implementation Details

We employed Isaac Gym [31] to collect data using 4,096 parallel environments, each with a Unitree Go1 robot [13] and a skateboard measuring $0.69 \times 0.27 \times 0.13\,\text{m}$ with a $0.43\,\text{m}$ wheelbase. We set the steering parameters in Eq. (1) to $\gamma_F^1 = 1.0$, $\gamma_F^2 = 1.12$, $\gamma_R^1 = 0.7$, and $\gamma_R^2 = 0.9$, yielding realistic behavior where a 10° deck roll induces about 11.0° front and 6.2° rear axle yaw. We tuned the front truck to favor responsiveness for a balance between stability and agility.

TABLE III. Reward Composition for Skateboard-Riding Skills: $\mathcal{R}_M = \mathcal{R}_{\text{dep}} + \mathcal{R}_{\text{ind}}$. (For a more detailed description, please refer to Sec. IV-B.)

Reward Term	**Formulation**	**Error Term** $\varepsilon_i (i = 0, 1, \cdots, 7)$	
		$M(\phi_t) =$ CARVING	$M(\phi_t) =$ PUSHING
Mode-Dependent Rewards $\mathcal{R}_{\text{dep}} = \sum_{i=1}^{7} r_i$			
r_1 : Command	$5.0 \exp(-\varepsilon_1/0.3)$	$\lvert c_\omega - \omega^{\mathcal{S}}_{D,z} \rvert$	$0.6 \lvert c_v - \mathrm{v}^{\mathcal{S}}_{D,x} \rvert + 0.2 \lvert \mathrm{v}^{\mathcal{S}}_{D,y} \rvert + 0.2 \lvert \omega^{\mathcal{S}}_{D,z} \rvert$
r_2 : Body Position	$2.0 \exp(-\varepsilon_2/0.2)$	$\lvert \mathbf{p}^{\mathcal{S}}_{B,xz} \rvert + 0.2 \lvert \mathrm{p}^{\mathcal{S}}_{B,y} \rvert + \lvert \mathrm{p}^{\mathcal{S}}_{B,z} - \mathrm{p}^{\text{carving}}_{B,z} \rvert$	$\lVert \mathbf{p}^{\mathcal{S}}_{B} - \mathbf{p}^{\text{pushing}}_{B} \rVert_2$
r_3 : Foot Position	$2.0 \exp(-\varepsilon_3/0.3)$	$\sum_{i \in \mathcal{F}_{\text{all}}} \lvert \mathbf{p}^{\mathcal{S}}_{F_i,xy} - \mathbf{p}^{\text{carving}}_{F_i,xy} \rvert$	$\sum_{i \in \mathcal{F}_{\text{right}}} \lvert \mathbf{p}^{\mathcal{S}}_{F_i,xy} - \mathbf{p}^{\text{pushing}}_{F_i,xy} \rvert + \sum_{j \in \mathcal{F}_{\text{left}}} 0.2 \lvert \mathrm{p}^{\mathcal{S}}_{F_j,x} - \mathrm{p}^{\text{pushing}}_{F_j,x} \rvert + 0.8 \lvert \mathrm{p}^{\mathcal{S}}_{F_j,y} - \mathrm{p}^{\text{pushing}}_{F_j,y} \rvert$
r_4 : Riding Posture	$2.0 \exp(-\varepsilon_4/0.4)$	$\lVert \boldsymbol{q} - \boldsymbol{q}^{\text{carving}} \rVert_1$	$0.7 \sum_{i \in \mathcal{J}_{\text{right}}} \lvert q_i - q^{\text{pushing}}_i \rvert + 0.3 \sum_{j \in \mathcal{J}_{\text{left}}} \lvert q_j - q^{\text{pushing}}_j \rvert$
r_5 : Foot Slip	$1.0 \exp(-\varepsilon_5/0.4)$	$\sum_{i \in \mathcal{F}_{\text{all}}} \lVert \mathbf{v}^{\mathcal{D}}_{F_i,xy} \rVert_2 \mathbb{1}_{\{c_i=1\}}$	$\sum_{i \in \mathcal{F}_{\text{right}}} \lVert \mathbf{v}^{\mathcal{D}}_{F_i,xy} \rVert_2 \mathbb{1}_{\{c_i=1\}} + \sum_{i \in \mathcal{F}_{\text{left}}} \lVert \mathbf{v}^{\mathcal{W}}_{F_i,xy} \rVert_2 \mathbb{1}_{\{c_i=1\}}$
r_6 : Contact Pattern	$-2.0\, \varepsilon_6$	$4 - \sum_{i \in \mathcal{F}_{\text{all}}} \mathbb{1}_{\{c_i=1\}}$	$2 - \sum_{i \in \mathcal{F}_{\text{right}}} \mathbb{1}_{\{c_i=1\}}$
r_7 : Foot Clearance	$-0.4\, \varepsilon_7$	0	$\sum_{i \in \mathcal{F}_{\text{left}}} \max(0.1 - p^{\mathcal{W}}_{F_i,z}, 0) \mathbb{1}_{\{c_i=0\}}$
Mode-Independent Rewards $\mathcal{R}_{\text{ind}} = \sum_{i=8}^{11} r_i$			
r_8 : Alignment	$2.0 \exp(-\lVert \boldsymbol{\theta}^{\mathcal{S}}_{B,xy} \rVert_2 / 0.5) + 2.0 \exp(-\lvert \theta^{\mathcal{S}}_{B,z} \rvert / 0.2) + \exp(-\lVert \mathbf{v}^{\mathcal{S}}_{B,xy} \rVert_2 / 0.3) + 2.0 \exp(-\sum_{i \in \mathcal{F}_{\text{all}}} \lVert \boldsymbol{\theta}^{\mathcal{S}}_{F_i,xz} \rVert / 0.9)$		
r_9 : Smoothness	$-5e\text{-}6 \lVert \mathbf{a} - \mathbf{a}_{t-1} \rVert_2 - 1e\text{-}6 \lVert \dot{\boldsymbol{q}} \rVert_2 - 1e\text{-}7 \lVert \ddot{\boldsymbol{q}} \rVert_2 - 4e\text{-}7 \lVert \boldsymbol{\tau}_{\boldsymbol{q}} \rVert_2 - 1e\text{-}6 \sum_{j \in \mathcal{J}_{\text{all}}} \max(\tau_q[j] \dot{q}[j], 0.0)$		
r_{10} : Stabilization	$-1e\text{-}6 \lVert \dot{\mathbf{v}}^{\mathcal{W}}_{B} \rVert_2 - 7e\text{-}2 \lVert \boldsymbol{\omega}^{\mathcal{B}}_{B,xy} \rVert_2 - 2.0 \lvert \mathrm{v}^{\mathcal{W}}_{B,z} \rvert - 1e\text{-}5 \lVert \dot{\mathbf{v}}^{\mathcal{W}}_{D} \rVert_2 - 1e\text{-}4 \lvert \omega^{\mathcal{S}}_{D,y} \rvert - 1e\text{-}4 \lVert \dot{\boldsymbol{\omega}}^{\mathcal{S}}_{D} \rVert_2$		
r_{11} : Safety	$-2.0 \sum_{i \in \mathcal{F}_{\text{all}}} \mathbb{1}_{\{F_i \text{ is on edge}\}} - 1.0 \sum_{i \in \mathcal{F}_{\text{all}}} \max(\lVert \mathbf{f}_{F_i} \rVert_2 - 80.0, 0.0) - 2.5\, \mathbb{1}_{\{\text{collision}\}} - 6.0\, \mathbb{1}_{\{\text{termination}\}} - 2.0\, \mathbb{1}_{\{\text{joint limit}\}}$		

To enhance domain adaptability, we randomized the intrinsic parameters $\mathbf{x}_{\text{int}}$ of both the robot and skateboard within the ranges listed in TABLE I. The overall architecture and network configuration are illustrated in Fig.4 and TABLE II. To improve robustness to external perturbations and sudden command changes, we applied random impulses to the body and deck every 3 s and resampled the command $\boldsymbol{c}$ every 5 s.

We used a stochastic policy π_θ that samples actions from a diagonal Gaussian, with the mean predicted by the actor network f^{actor}_θ and learnable standard deviations $\boldsymbol{\theta}_{\text{std}} \in \mathbb{R}^{12}$. The policy converged in about 10 h on an RTX 4090 GPU with an Intel i9-9900K CPU, followed by 5 h for distillation.

V. EXPERIMENTAL RESULTS

We demonstrate the effectiveness of the proposed Phase-Aware Policy Learning (*PAPL*) framework through a series of experiments. First, we verify the intrinsic multi-modality of skateboarding motions, motivating the adoption of a phase-aware modulation. Next, we evaluate the command-tracking accuracy and conduct ablation studies to quantify the contribution of individual components. Finally, we compare energy efficiency with other locomotion baselines and demonstrate sim-to-real transferability on a physical skateboard platform.

A. Multi-Modality of Skateboarding Motions

Skateboarding inherently involves multi-modal behaviors across distinct phases. To verify this property, we visualized the distributions of action, observation, and critic-value data collected over 80 s of execution using a proficient skateboard-riding policy. To collect a range of motion trajectories, we resampled the command $\boldsymbol{c}$ every 10 s within $c_v \in [0.0, 1.5]$ and $c_w \in [-0.5, 0.5]$, with a fixed phase period of $T_\phi = 10$ s.

As illustrated in Fig. 6, t-SNE projections of action and observation data show mode-dependent clustering: PUSHING (red) and CARVING (blue) occupy separated regions, while TRANSITION (green) lies in between. While dimensionality

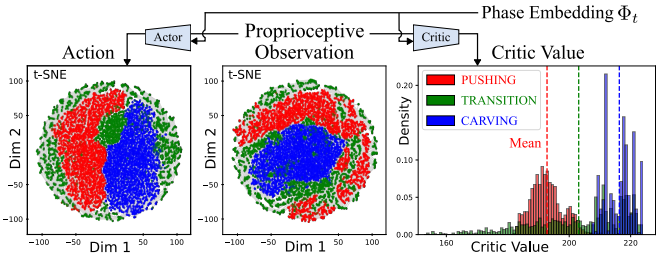

Fig. 6. Visualization of action $\boldsymbol{a}$, proprioceptive observation $\mathbf{o}_{\text{prop}}$, and critic-value V distributions over 80 s of skateboard riding. Action and observation data are projected using t-SNE [32], and critic values are visualized as histograms. Data points are color-coded by the corresponding mode $M(\phi_t)$ at the time of collection. Further details are provided in Sec. V-A.

reduction could exaggerate separation, the observed structure indicates that the actor needs to address phase-specific variations in its input–output distributions. On the critic side, the value histogram also exhibits distinct distributions reflecting mode-dependent rewards $\mathcal{R}_M$, listed in TABLE III. These findings motivate the use of phase-conditioned modulation to effectively address different modalities using a unified policy.

B. Evaluation of Skateboarding Performance

We evaluated skateboarding performance and the contribution of each *PAPL* component by measuring time-averaged command-tracking errors of our policy and its ablated variants. We defined the error term as $\lvert c_v - v^{\mathcal{S}}_{D,x} \rvert \mathbb{1}_{\{M=\text{PUSHING}\}} + \lvert c_\omega - \omega^{\mathcal{S}}_{D,z} \rvert \mathbb{1}_{\{M=\text{CARVING}\}}$ and computed it over the command set sampled from $c_v \in [0.0, 1.5]$ and $c_\omega \in [-0.5, 0.5]$ at 0.05 resolution. For each command, ten environments were initialized with intrinsic properties drawn from the test ranges in TABLE I. Robots executed c_v for 5 s and c_ω for another 5 s ($T_\phi = 10$ s), repeated over a 40 s horizon, while random body perturbations are applied every 4 s to assess robustness.

Fig. 7 shows tracking-error heatmaps and corresponding command-area curves [15], where the command area denotes the portion of the command space within a specified error

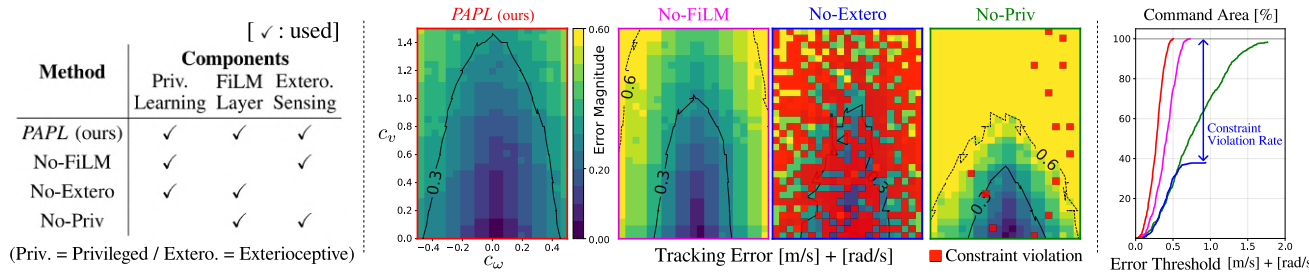

Fig. 7. **Left**: Component configurations of the proposed *PAPL* framework and ablated variants. **Middle**: Tracking error heatmaps with contours, where darker regions indicate lower errors. Contours represent iso-error boundaries, while red regions denote constraint violations when the robot either overturned or deviated more than $0.5\,\mathrm{m}$ from the board. **Right**: Command-area curves showing the percentage of commands tracked within given error thresholds.

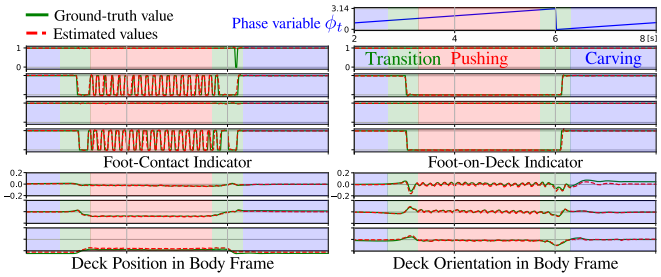

Fig. 8. Estimation performance of the extrinsic estimator across skateboarding phases. It accurately infers foot-contact and foot-on-deck indicators, as well as deck position and orientation in the body frame, closely matching ground truth values during skateboarding at $c_v = 0.8\,\mathrm{m/s}$ with $T_\phi = 6\,\mathrm{s}$.

threshold. Trials in which the robot overturned or deviated more than $0.5\,\mathrm{m}$ from the board were treated as constraint violations and excluded from the command-area calculation. *PAPL* presents the largest region with tracking error below 0.3 and no constraint violations, demonstrating robustness across broad command spaces and under external disturbances. By contrast, the No-FiLM variant, which employs standard MLP backbones for both actor and critic, exhibits a narrower region with higher errors. This empirically demonstrates that the inductive bias introduced by the FiLM architecture is crucial to address the multi-modality inherent in skateboarding motions. Meanwhile, the No-Extero variant incurs frequent constraint violations, underscoring the critical role of exteroceptive sensing. In particular, the policy often fails to correct external perturbations or persistent drifts, which eventually leads to collapse. The No-Priv variant fails to acquire effective motions and, in most cases, converges to a standing-still behavior to avoid penalties. We also evaluated the Mixture-of-Experts (MoE) MLP as the actor backbone in place of the FiLM MLP, but it failed to achieve effective skateboarding behaviors as the pushing-phase expert collapsed without producing meaningful actions. This failure was likely due to phase-specific isolation, limited information sharing among experts, and the large parameter count, which under limited GPU memory restricted training to fewer environments, reducing data collection and exploration efficiency and leading to trivial behaviors such as remaining inactive.

The command-area curves show that *PAPL* attains broader command-space coverage under strict error thresholds, highlighting the complementary roles of its components. Fig. 8 further shows the extrinsic estimator's accuracy in inferring privileged extrinsic states $\mathbf{x}_{\text{ext}}$ from observation histories; additional results are available in the supplementary video.

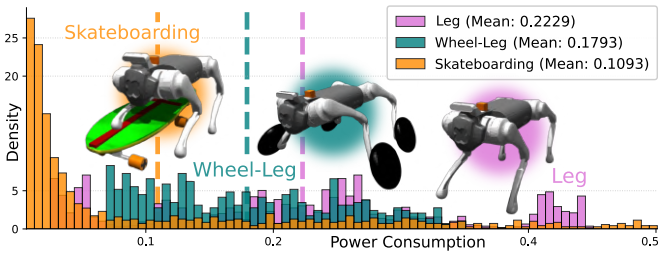

Fig. 9. Consumed motor power distributions for three locomotion strategies over a $30\,\mathrm{m}$ traversal. Vertical dashed lines denote the mean values.

C. Power Consumption Analysis

To compare locomotion efficiency, we measured normalized motor power [27] required to reach a target $30\,\mathrm{m}$ ahead on flat ground. For the wheel-legged and legged baselines, the forward velocity command was fixed at $1.5\,\mathrm{m/s}$, and heading was stabilized using yaw-rate commands from a PD controller on heading error. For the skateboarding, we strategically modulated the phase clock ϕ_t: propulsion was activated only when the forward velocity dropped below $0.7\,\mathrm{m/s}$, otherwise the clock was halted in the middle of carving phases to focus solely on steering. Although this comparison was conducted on flat terrain with moderate friction favorable to skateboarding, Fig. 9 shows that skateboarding consumes less power than the other two locomotion strategies. The histogram of skateboarding exhibits a long-tail distribution, reflecting its unique characteristics: steering maneuvers require relatively little energy, whereas motions such as pushing off and mounting need much higher power.

D. Real-World Experiments

We applied the proposed Phase-Aware Policy Learning (*PAPL*) framework to a Unitree Go1 robot riding a physical skateboard to examine its real-world applicability. As shown in Fig. 10-(a), the robot reproduced pushing, transition, and carving behaviors via zero-shot transfer from simulation. We further conducted trials under diverse conditions, including external perturbations, low-light environments, and uneven sidewalks, as shown in Fig. 10-(b), confirming that the policy can be deployed without collapse. These experiments show the feasibility of our approach in hardware; for an intuitive understanding, please refer to the supplementary video.

VI. Conclusion

In this work, we introduced the Phase-Aware Policy Learning (*PAPL*) framework to endow quadruped robots with

(a) Real-world Skateboarding Snapshots

(b) Various Environmental Conditions

Fig. 10. Real-world demonstrations of quadruped skateboarding. (a) Snapshots captured at different modes. (b) Various experimental conditions: external perturbation, low-light settings, and uneven sidewalks.

skateboard-riding capability for efficient transportation. By integrating a phase-conditioned RL formulation with FiLM-based layer modulation, asymmetric privileged learning, and exteroceptive sensing, *PAPL* achieves robust and agile skateboarding behaviors, including steering via deck tilting. Simulation results demonstrated moderate command tracking performance of robots using skateboards, improved energy efficiency over other locomotion baselines on the flat ground, and the complementary role of each component through ablation studies. Real-world experiments further demonstrated zero-shot transfer from simulation, where the robot successfully operated under diverse conditions including external perturbations, low-light settings, and uneven terrains.

As future work, we plan to integrate a high-level navigation policy that generates velocity commands and modulates the phase clock in response to surrounding environmental conditions—particularly for timely mounting and propulsion—and to incorporate front-facing perception modules for velocity estimation and navigation command generation.

Acknowledgments

This work was supported by the Korea government (MSIT) through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants (RS-2025-25443318 and RS-2023-00237965) and the National Research Foundation of Korea (NRF) grant (RS-2023-00208506).

References

[1] P. Arm et al., "Scientific exploration of challenging planetary analog environments with a team of legged robots", *Science Robotics*, vol. 8, no. 80, pp. eade9548, 2023.

[2] B. Lindqvist et al., "Multimodality robotic systems: Integrated combined legged-aerial mobility for subterranean search-and-rescue", *Robotics and Autonomous Systems*, vol. 154, pp. 104134, 2022.

[3] C. D. Bellicoso et al., "Advances in real-world applications for legged robots", *Field Robotics*, vol. 35, no. 8, pp. 1311–1326, 2018.

[4] J. Hwangbo et al., "Raibo2: Highly efficient quadruped robot completing full marathon with a single battery charge", 2025.

[5] W. Thibault et al., "Learning skateboarding for humanoid robots through massively parallel reinforcement learning", *arXiv preprint arXiv:2409.07846*, 2024.

[6] N. Takasugi et al., "Extended three-dimensional walking and skating motion generation for multiple noncoplanar contacts with anisotropic friction: Application to walk and skateboard and roller skate", *IEEE Robotics and Automation Letters (RA-L)*, vol. 4, no. 1, pp. 9–16, 2018.

[7] K. Kimura et al., "Riding and speed governing for parallel two-wheeled scooter based on sequential online learning control by humanoid robot", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[8] J. Anglingdarma et al., "Motion planning and feedback control for bipedal robots riding a snakeboard", in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 2818–2824.

[9] S. Xin et al., "A torque-controlled humanoid robot riding on a two-wheeled mobile platform", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1435–1442.

[10] V. Rajendran et al., "Towards humanoids using personal transporters: Learning to ride a segway from humans", in *IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics*. IEEE, 2022, pp. 01–08.

[11] Z. Xu et al., "Optimization based dynamic skateboarding of quadrupedal robot", in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 8058–8064.

[12] H. Liu et al., "Discrete-time hybrid automata learning: Legged locomotion meets skateboarding", *arXiv preprint arXiv:2503.01842*, 2025.

[13] Unitree, "Go1", `https://www.unitree.com/go1`, 2021, Accessed: 2024-11-27.

[14] M. Rosatello et al., "The skateboard speed wobble", in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2015, vol. 57168, p. V006T10A054.

[15] G. B. Margolis et al., "Rapid locomotion via reinforcement learning", *The International Journal of Robotics Research (IJRR)*, vol. 43, no. 4, pp. 572–587, 2024.

[16] J. Schulman et al., "Proximal policy optimization algorithms", *arXiv preprint arXiv:1707.06347*, 2017.

[17] S. Ross et al., "A reduction of imitation learning and structured prediction to no-regret online learning", in *International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[18] T. Gangwani et al., "Learning belief representations for imitation learning in pomdps", in *Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 1061–1071.

[19] L. Meng et al., "Memory-based deep reinforcement learning for pomdps", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5619–5626.

[20] Z. Fu et al., "Deep whole-body control: learning a unified policy for manipulation and locomotion", in *Conference on Robot Learning (CoRL)*. PMLR, 2023, pp. 138–149.

[21] W. Yu et al., "Preparing for the unknown: Learning a universal policy with online system identification", in *Robotics: Science and Systems*, 2017.

[22] J. Lee et al., "Learning quadrupedal locomotion over challenging terrain", *Science Robotics*, vol. 5, no. 47, pp. eabc5986, 2020.

[23] T. Miki et al., "Learning robust perceptive locomotion for quadrupedal robots in the wild", *Science Robotics*, vol. 7, no. 62, pp. eabk2822, 2022.

[24] X. Cheng et al., "Extreme parkour with legged robots", in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 11443–11450.

[25] A. Kumar et al., "Rma: Rapid motor adaptation for legged robots", in *Robotics: Science and Systems*, 2021.

[26] A. Kumar et al., "Adapting rapid motor adaptation for bipedal robots", in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1161–1168.

[27] M. Yoon et al., "Enhancing navigation efficiency of quadruped robots via leveraging personal transportation platforms", in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 11184–11190.

[28] E. Perez et al., "Film: Visual reasoning with a general conditioning layer", in *AAAI conference on artificial intelligence*, 2018, vol. 32.

[29] L. Bauersfeld et al., "User-conditioned neural control policies for mobile robotics", in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1342–1348.

[30] C. Chi et al., "Diffusion policy: Visuomotor policy learning via action diffusion", *The International Journal of Robotics Research (IJRR)*, p. 02783649241273668, 2023.

[31] V. Makoviychuk et al., "Isaac gym: High performance gpu-based physics simulation for robot learning", *arXiv preprint arXiv:2108.10470*, 2021.

[32] L. Maaten et al., "Visualizing data using t-sne", *Journal of Machine Learning Research (JMLR)*, vol. 9, no. Nov, pp. 2579–2605, 2008.