# Learning Forward Looking Adaptation to Dynamic Payloads for Quadruped Locomotion via Physics-Informed Neural Networks

Author Names Omitted for Anonymous Review. Paper-ID [add your ID here]

***Abstract*— Payload adaptive locomotion an essential skill for quadruped robots deployed in many real world scenarios, particularly when tasked with transporting a dynamic payload. Existing methods face fundamental limitations: reactive adaptation strategies respond too slowly to sudden payload shifts, while learning-based approaches often produce physically inconsistent models of the robot's dynamics that generalize poorly to novel states. To address these limitations, we propose Forward Looking Adaption to Dynamic Payloads (FLAP), a novel method that proactively compensates for discrepancies between expected and actual locomotion behavior caused by dynamic payloads. FLAP integrates two key components: (1) a novel physics-informed neural network (PINN) that learns to predict anticipated joint states while enforcing physical consistency through structured dynamics constraints, and (2) a composite adaptive control law that rapidly generates anticipatory joint torque compensations using the PINN's predictions. Our approach ensures generalizable dynamics modeling while enabling real-time, forward-looking adaptation to dynamic payload conditions. Experimental results demonstrate that FLAP effectively enables robust locomotion with dynamic payloads in real-world environments.**

## I. Introduction

Quadruped robots have demonstrated remarkable advances in mobility driven by their potential to operate in unstructured environments [1]–[9]. These capabilities have enabled their use in applications such as autonomous inspection [10]–[13], disaster response [14], [15], and search-and-rescue [16], [17]. In these scenarios, legged robots may be required to transport dynamic payloads such as shifting rubble, loosely packed containers, or liquids. Dynamic payloads introduce time-varying mass distributions that generate forces and torques that act on the robot's body. To maintain stable locomotion, the robot must rapidly adapt to these disturbances by compensating for any deviations in real time.

The challenge of payload-adaptive legged locomotion has motivated the development of diverse approaches. Model-based methods leverage control-theoretic principles and explicit knowledge of system dynamics [18]–[21] yet often exhibit slow adaptation to abrupt payload changes and require significant manual tuning. Recent learning-based approaches [5], [6], [22]–[30] have demonstrated promising results, but current state-of-the-art methods frequently fail to learn physically-consistent representations of the robot's dynamics. This limitation leads to substantial performance degradation when encountering novel states outside the training distribution, highlighting a critical gap in robust generalization for real-world deployment with dynamic payloads.

We present the ***F**orward **L**ooking **A**daptation to Dynamic **P**ayloads* (FLAP) method (Fig. 1) for real-time adaptation of quadruped locomotion controllers to dynamic payloads. FLAP enables proactive compensation for anticipated locomotion discrepancies caused by shifting payload mass distributions through a physics-informed neural network (PINN) that models the system's forward dynamics. The PINN jointly predicts the evolution of the robot's whole body state and the payload induced forces/torques acting on the robot, while a physics-informed loss function ensures the predictions respect the known dynamics of the robot. This constrained learning approach results in physically plausible forward dynamics model that generalizes well to novel states. During deployment, these learned dynamics are integrated into a composite adaptive control law that rapidly generates anticipatory joint torque compensations in real time. By combining physics informed learning with adaptive control, FLAP achieves robust locomotion with adaption to dynamic payloads.

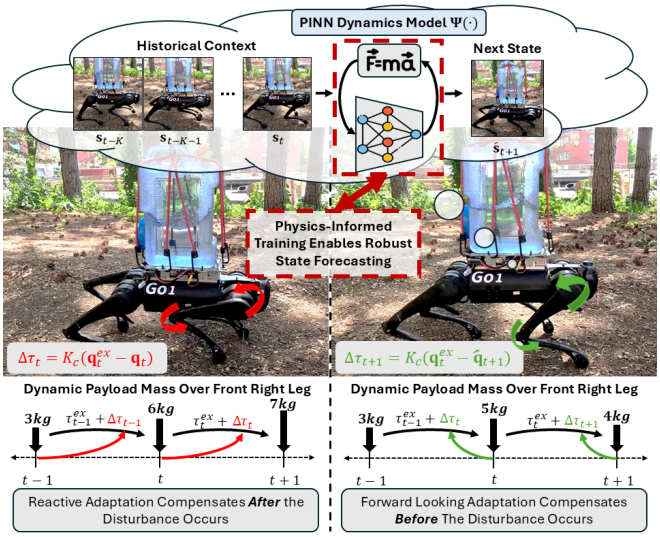


Fig. 1. Transporting dynamic payloads (e.g., liquids) is challenging due to time-varying mass distributions that overwhelm reactive control. Our physics-informed framework enables accurate state forecasting for "forward looking" proactive adaptation, achieving robust real-world transport.

The main contribution of this work is the introduction of our novel learning-based FLAP method to address dynamic payload adaption for quadruped robots. Two specific novelties include:

- We enable rapid "forward looking" proactive adaption to the changing mass distribution of dynamic payloads being transported by quadruped robots.
- We introduce a novel physics informed neural network for modeling quadruped robot dynamics enabling greater generalization to novel system states during deployment.

## II. Related Work

Current approaches to payload adaptive quadruped locomotion can generally be split into learning and non-learning based approaches.

### A. Non-Learning Approaches for Payload Adaptation

Non-learning approaches to adaptive quadruped locomotion typically rely on analytical dynamics modeling and control-theoretic frameworks. Early methods employed online payload identification techniques [18], [31] or force-based quadratic programming [19]. Subsequent advances introduced stability-guaranteed formulations using Control Lyapunov Functions (CLFs) [20] to ensure energy dissipation toward desired states. Recent work has integrated L1 adaptive control [32] with model predictive control (MPC) to achieve robust adaptation [21], while others have leveraged template models [33] to simplify nonlinear dynamics coupled with gradient-based adaptive laws [31]. While theoretically principled, these methods often require complex control architectures with substantial computational overhead and significant domain expertise to implement effectively.

### B. Learning Approaches for Payload Adaptation

Current approaches for learning adaptive quadruped locomotion employ reinforcement learning (RL) with varying adaptation strategies. Privileged learning methods [6], [22]–[25], [34], [35] leverage "privileged" state data (e.g., payload mass, terrain properties) only available in simulation to learn adaptive policies, but generalize poorly beyond training conditions [36]. Online learning techniques [26]–[30] address this through in-situ training, though their slow convergence limits responsiveness to sudden payload changes. Hybrid approaches combine RL with classical adaptive control, either by modifying MPC inputs/outputs [37], [38] or through complementary adaptive controllers [39], but remain constrained to learned configurations. Recent work explores hierarchical policies [40], [41], that first learn a nominal locomotion policy followed by an complimentary adaptive policy for generating compensation values. However, these methods all share a critical limitation: they neglect the underlying differential equations governing robot dynamics, resulting in physically inconsistent models that impair generalization.

### C. Physics Informed Neural Networks (PINNs)

Physics-informed neural networks (PINNs) embed governing differential equations directly into their training objective, enforcing physical consistency via equation-based loss terms leveraging automatic differentiation tools [42]. While widely applied in scientific computing problems such as fluid dynamics [43]–[45], heat transfer [46], [47], and solid mechanics [48]–[51], PINNs have recently gained traction in robotics [52]. Applications include modeling manipulator dynamics [53], [54], quadrotor dynamics [55], [56], soft robots [57]–[60], state estimation [61]–[63], and motion planning [64], [65], demonstrating their potential for learning physically accurate representations of robotic systems.

## III. Approach

**Notation:** Matrices are denoted as boldface uppercase letters (e.g. $\mathbf{M} = \{M_{i,j}\} \in \mathbb{R}^{n\times m}$ is an $n \times m$ matrix whose element in the $i$-th row and $j$-th column is $M_{i,j}$), vectors are denoted as boldface lowercase letters (e.g., $\mathbf{v} \in \mathbb{R}^d$ is a $d$-dimensional vector whose $i$-th element is $v_i$), and scalars are represented by non-boldface characters (e.g., $s$, $\alpha$).

### A. Forward Looking Dynamic Payload Compensation Problem Formulation

During locomotion, a quadruped robot carrying a payload gathers proprioceptive observations through its onboard sensors. Joint encoders provide leg positions $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$, while an IMU measures torso orientation $\Theta$, angular velocity $\dot{\Theta}$, and linear acceleration $\ddot{\mathbf{p}}$. A Kalman filter estimates the global torso position $\mathbf{p}$ and velocity $\dot{\mathbf{p}}$ from kinematic data and IMU measurements. The system also computes exerted joint torques $\boldsymbol{\tau}$ using $\dot{\mathbf{q}}$, motor current, and models of the joint actuators. At each timestep, proprioceptive observations are aggregated into a state vector $\mathbf{s} = [\dot{\mathbf{p}}, \Theta, \dot{\Theta}, \mathbf{q}, \dot{\mathbf{q}}]$, which serves as input to the locomotion controller.

The expected locomotion behavior is defined by expected torso velocities $\mathbf{b}^{ex} = [\dot{\mathbf{p}}^{ex}, \dot{\Theta}^{ex}]^\top$, typically provided by a motion planner or user input. A parameterized controller $\Phi(\cdot)$ then uses $\mathbf{s}$ and $\mathbf{b}^{ex}$ to generate joint position and torque commands $\Phi : (\mathbf{s}, \mathbf{b}^{ex}) \mapsto (\mathbf{q}^{ex}, \boldsymbol{\tau}^{ex})$. $\Phi(\cdot)$ encodes the quadruped robot's dynamics which can be expressed as:

$$\begin{pmatrix}\mathbf{0}_6 \\ \boldsymbol{\tau}\end{pmatrix} + \mathbf{J}_c^\top \mathbf{f}^r = \mathbf{A}\ddot{\boldsymbol{\rho}} + \mathbf{c} + \mathbf{g} \tag{1}$$

where $\ddot{\boldsymbol{\rho}} = [\ddot{\mathbf{p}}, \ddot{\Theta}, \ddot{\mathbf{q}}]^\top$, $\mathbf{A}$ (generalized mass matrix), $\mathbf{c}$ (Coriolis forces), $\mathbf{g}$ (gravitational forces), $\mathbf{J}_c^\top$ (contact Jacobian), and $\mathbf{f}^r$ (ground reaction forces) characterize the system dynamics. The null vector $\mathbf{0}_6$ corresponds to the torso's unactuated degrees of freedom. Eq. 1 represents the second-order dynamics relating robot accelerations to applied forces/torques.

When transporting a payload, the additional mass shifts the system dynamics away from Eq. 1. This manifests primarily as joint torque disturbances $\Delta\boldsymbol{\tau} = \mathbf{J}_c^\top \Delta\mathbf{f}^r$, where $\Delta\mathbf{f}^r$ represents unmodeled variations in ground reaction forces caused by the mass of the payload. The modified dynamics now take the form:

$$\begin{pmatrix}\mathbf{0}_6 \\ \boldsymbol{\tau}\end{pmatrix} + \mathbf{J}_c^\top \mathbf{f}^r + \Delta\boldsymbol{\tau} = \mathbf{A}\ddot{\boldsymbol{\rho}} + \mathbf{c} + \mathbf{g} \tag{2}$$

This model mismatch causes $\mathbf{q}$ to deviate from $\mathbf{q}^{ex}$, resulting in position errors $\Delta\mathbf{q} = \mathbf{q}^{ex} - \mathbf{q} \neq \mathbf{0}$ that degrade locomotion performance.

In order to account for the $\Delta\boldsymbol{\tau}$ introduced by the payload, we compensate $\Phi(\cdot)$ with joint torque offsets:

$$\boldsymbol{\tau}^{ex} = \Phi(\mathbf{s}, \mathbf{b}^{ex}) + \Delta\boldsymbol{\tau} \tag{3}$$

While direct measurement of $\Delta\boldsymbol{\tau}$ is infeasible without foot force sensors, we leverage adaptive control principles [39], [66], [67] to estimate it via joint position errors:

$$\Delta\boldsymbol{\tau} \approx \mathbf{K}_c \Delta\mathbf{q} = \mathbf{K}_c(\mathbf{q}^{ex} - \mathbf{q}) \tag{4}$$

where $\mathbf{K}_c$ is an adaptive gain matrix. As a result, the compensated torque command explicitly accounts for unmodeled dynamics through measurable state deviations.

This formulation follows a "reactive" adaptive control paradigm, where state feedback compensates for disturbances in real time, a common approach for dynamic environments. However, reactive control lacks foresight: the commanded joint state $\mathbf{q}^{ex}$ targets the next timestep, while $\mathbf{q}$ reflects only the current state. Since compensation via $\Delta\mathbf{q}$ relies solely on instantaneous sensory inputs, it cannot preemptively counteract impending disturbances from a dynamic payload. To address this, we propose a proactive approach using a learned forward dynamics model $\Omega(\cdot)$. By processing a history of $K$ expected behavior, state, and previous-action triplets $\mathbf{X}_t = [\mathbf{x}_{t-K}, \ldots, \mathbf{x}_t]^\top$, where $\mathbf{x}_i = [\mathbf{b}_i^{ex}, \mathbf{s}_i, \mathbf{q}_{i-1}^{ex}]^\top$, $\Omega(\cdot)$ predicts the future joint state $\hat{\mathbf{q}}_{t+1} = \Omega(\mathbf{X}_t)$. This enables anticipatory compensation via:

$$\Delta\boldsymbol{\tau}_{t+1} = \mathbf{K}_c(\mathbf{q}_t^{ex} - \hat{\mathbf{q}}_{t+1}) \tag{5}$$

effectively transforming the adaptation procedure from reactive to proactive, enabling "forward looking" adaptive locomotion for dynamic payloads.

The problem of learning $\Omega(\cdot)$ can be formulated as:

$$\underset{\Omega}{\operatorname{argmin}} \sum_{(\mathbf{X}_i, \mathbf{q}_{i+1}) \in \mathbf{D}} \|\mathbf{q}_{i+1} - \Omega(\mathbf{X}_i)\|^2 \tag{6}$$

where $\mathbf{D}$ contains $N$ samples across varied payloads and locomotion behaviors. This standard regression formulation trains $\Omega(\cdot)$ to minimize the next-step joint prediction error to enable proactive compensation (Eq. 5). However, two key limitations arise: (1) the physics-agnostic approach lacks explicit incorporation of the robot's physical dynamics [68], resulting in (2) exhaustive training data requirements to avoid overfitting [6], [23], [24]. Consequently, without sufficient coverage, the model generalizes poorly to unseen states during deployment.

### *B. PINN for Learning Forward Looking Adaptation to Dynamic Payloads*

We present a novel physics informed neural network approach for learning FLAP while addressing the aforementioned limitations. Our approach trains a model $\Psi : \mathbf{X}_t \mapsto (\bar{\mathbf{s}}_{t+1}, \boldsymbol{\tau}_{t+1}^{net})$ to jointly predict the robot's future state and the net forces/torques acting on the robot. The state prediction $\bar{\mathbf{s}}_{t+1} = [\mathbf{p}_{t+1}^z, \Theta_{t+1}, \mathbf{q}_{t+1}]^\top$ captures torso height, orientation, and joint positions (excluding torso $x, y$ position for generalizability), while the force/torque prediction $\boldsymbol{\tau}_t^{net} = [\mathbf{f}_{t+1}^b, \boldsymbol{\tau}_{t+1}^b, \boldsymbol{\tau}_{t+1}]^\top + \mathbf{J}_c^\top \mathbf{f}_{t+1}^r$ includes torso forces $\mathbf{f}^b$, torques $\boldsymbol{\tau}^b$, and payload induced ground reaction forces $\mathbf{J}_c^\top \mathbf{f}^r$ (available as privileged simulation data). Through this formulation, $\Psi(\cdot)$ explicitly learns to model ground reaction force disturbances ($\mathbf{J}_c^\top \Delta\mathbf{f}^r$) while capturing the coupled robot-payload dynamics governing state evolution.

While jointly predicting $\bar{\mathbf{s}}_{t+1}$ and $\boldsymbol{\tau}_{t+1}^{net}$ captures coupled dynamics, this naive approach lacks explicit physical constraints, risking overfitting to $\mathbf{D}$ without learning a physically consistent representation of the robot's dynamics. We address

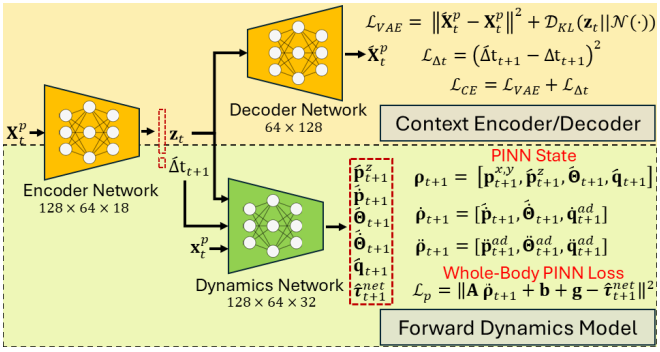


Fig. 2. **PINN Architecture**: The context encoder processes historical data to produce a latent representation and timestep prediction, while the forward dynamics model uses these outputs to predict future states and forces/torques via physics informed learning.

this by formulating $\Psi(\cdot)$ as a Physics Informed Neural Network (PINN) [42], decomposed into two jointly-trained components as seen in Fig. 2. A context encoder/decoder ($\Psi_{ENC}/\Psi_{DEC}$) processing historical system states and a forward dynamics model ($\Psi_{FD}$) predicting future states and forces

The context encoder, $\Psi_{ENC} : \mathbf{X}_t^p \mapsto (\mathbf{z}_t, \hat{\Delta}t_{t+1})$, processes a time-augmented history $\mathbf{X}_t^p = [\mathbf{x}_{t-K}^p, \ldots, \mathbf{x}_t^p]^\top$, where each $\mathbf{x}_i^p = [\Delta t_i, \mathbf{b}_i^{ex}, \mathbf{s}_i, \mathbf{q}_{i-1}^{ex}]^\top$ contains the timestep $\Delta t_i$ between action $\mathbf{q}_{i-1}^{ex}$ and resulting state $\mathbf{s}_i$. The encoder outputs a latent dynamics representation $\mathbf{z}_t$ and predicts the next timestep $\hat{\Delta}t_{t+1}$, while a decoder, $\Psi_{DEC} : \mathbf{z}_t \mapsto \hat{\mathbf{X}}_t^p$, reconstructs $\mathbf{X}_t^p$ from $\mathbf{z}_t$ to ensure it preserves essential features. The forward dynamics model, $\Psi_{FD} : (\mathbf{z}_t, \hat{\Delta}t_t, \mathbf{x}_t^p) \mapsto (\bar{\mathbf{s}}_{t+1}, \boldsymbol{\tau}_{t+1}^{net})$ then uses $\mathbf{z}_t$, $\hat{\Delta}t_{t+1}$, and $\mathbf{x}_t^p$ to predict the next state $\bar{\mathbf{s}}_{t+1}$ and net forces/torques $\boldsymbol{\tau}_{t+1}^{net}$, explicitly modeling the time-dependent system evolution in Eq. 2.

We ensure $\Psi(\cdot)$ learns a physically consistent representation of the robot's dynamics by constructing loss functions that enforce learning accurate higher-order behaviors (e.g. system velocities and accelerations). The higher-order representations learned by $\Psi(\cdot)$ can be derived using the automatic differentiation (AD) tools implemented in deep learning programming frameworks [69]–[71]. For joint dynamics, we calculate the learned relationship between position and velocity as $\dot{\mathbf{q}}_{t+1}^{ad} = f_{ad}(\hat{\Delta}t_{t+1}, \hat{\mathbf{q}}_{t+1})$ via AD and minimize:

$$\mathcal{L}_{\dot{\mathbf{q}}_{t+1}}^p = \|\dot{\mathbf{q}}_{t+1} - \dot{\mathbf{q}}_{t+1}^{ad}\|^2 \tag{7}$$

where $f_{ad}(x, y) = \frac{\partial}{\partial x} y$. $\mathcal{L}_{\dot{\mathbf{q}}_{t+1}}^p$ explicitly encourages $\Psi(\cdot)$ to learn a physically consistent relationship between the position and velocity of the next timestep's leg joints that aligns with the system's known dynamics. Due to the $x, y$ position of the torso being excluded from the learning objective, we directly predict $\dot{\mathbf{p}}_{t+1}$ and also $\dot{\Theta}_{t+1}$ due to their tight coupling, yielding the complete state prediction target $\mathbf{s}_{t+1}^p = [\mathbf{p}_{t+1}^z, \Theta_{t+1}, \mathbf{q}_{t+1}, \dot{\mathbf{p}}_{t+1}, \dot{\Theta}_{t+1}]^\top$.

To enforce $\Psi(\cdot)$'s consistency with the dynamics in Eq. 2, we further compute second-order derivatives via AD. This results in joint accelerations $\ddot{\mathbf{q}}^{ad} = f_{ad}(\hat{\Delta}t, \hat{\dot{\mathbf{q}}}^{ad})$, and torso linear/angular accelerations $\ddot{\mathbf{p}}_{t+1}^{ad} = f_{ad}(\hat{\Delta}t_t, \hat{\dot{\mathbf{p}}}_{t+1})$, $\ddot{\Theta}_{t+1}^{ad} = f_{ad}(\hat{\Delta}t_t, \hat{\dot{\Theta}}_{t+1})$. Using the predicted state and approximated

higher-order derivatives, the PINN whole body dynamics state of the system can be constructed as follows. The position is defined as $\boldsymbol{\rho}^p_{t+1} = [\mathbf{p}^x_{t+1}, \mathbf{p}^y_{t+1}, \hat{\mathbf{p}}^z_{t+1}, \hat{\Theta}_{t+1}, \hat{\mathbf{q}}_{t+1}]^\top$ (with the torso's $x, y$ position copied from ground truth), with the velocities being $\dot{\boldsymbol{\rho}}^p_{t+1} = [\hat{\dot{\mathbf{p}}}_{t+1}, \hat{\dot{\Theta}}_{t+1}, \dot{\mathbf{q}}^{ad}_{t+1}]^\top$, and accelerations defined as $\ddot{\boldsymbol{\rho}}^p_{t+1} = [\ddot{\mathbf{p}}^{ad}_{t+1}, \ddot{\Theta}^{ad}_{t+1}, \ddot{\mathbf{q}}^{ad}_{t+1}]^\top$. The PINN state is then used to enforce learned consistency with the whole-body dynamics in Eq. 2 via the loss function:

$$\mathcal{L}^p_{wb} = \|\mathbf{A}\ddot{\boldsymbol{\rho}}^p_{t+1} + \mathbf{b} + \mathbf{g} - \hat{\boldsymbol{\tau}}^{net}_{t+1}\|^2 \tag{8}$$

where $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{g}$ are calculated using $\boldsymbol{\rho}^p_{t+1}$ and $\dot{\boldsymbol{\rho}}^p_{t+1}$. $\mathcal{L}^p_{wb}$ explicitly couples state predictions ($\hat{\mathbf{s}}^p_{t+1}$) and force/torque estimates ($\hat{\boldsymbol{\tau}}^{net}_{t+1}$) through the system dynamics, ensuring physical consistency.

Putting all these elements together we arrive in the final physics informed loss function:

$$\mathcal{L}_{\text{PINN}} = \mathcal{L}_{CE} + \mathcal{L}^p_n + \mathcal{L}^p_{wb} + \mathcal{L}^p_{\dot{\mathbf{q}}_{t+1}} \tag{9}$$

$\mathcal{L}_{\text{PINN}}$ combines four key loss components: the context encoder-decoder loss $\mathcal{L}_{CE}$, the next-state prediction loss $\mathcal{L}^p_n$, the whole-body dynamics loss $\mathcal{L}^p_{wb}$, and the joint velocity loss $\mathcal{L}^p_{jv}$. The context loss $\mathcal{L}_{CE} = \|\mathbf{X}^p_t - \hat{\mathbf{X}}^p_t\|^2 + D_{KL}(\mathbf{z}_t|\mathcal{N}(0,1)) + (\Delta t_{t+1} - \hat{\Delta} t_{t+1})^2$ simultaneously optimizes for accurate sequence reconstruction, latent space regularization, and timestep prediction. The prediction loss $\mathcal{L}^p_n = \|\mathbf{s}^p_{t+1} - \hat{\mathbf{s}}^p_{t+1}\|^2 + \|\boldsymbol{\tau}^{net}_{t+1} - \hat{\boldsymbol{\tau}}^{net}_{t+1}\|^2$ ensures precise forecasting of both the robot's state and net forces/torques. Physical consistency is enforced through $\mathcal{L}^p_{wb}$ (defined in Eq. 8) which incorporates the full system dynamics, along with $\mathcal{L}^p_{\dot{\mathbf{q}}_{t+1}}$ (from Eq. 7) that specifically maintains proper joint velocity relationships. Together, these loss terms enable $\Psi(\cdot)$ to learn physically consistent predictions while effectively summarizing historical observations. Complete implementation details of the training procedure are available in the supplementary material.

### C. Online Forward Looking Adaptive Control

The trained model $\Psi(\cdot)$ is incorporated into the adaptive control framework in Eq. 3 through:

$$\boldsymbol{\tau}^{ex}_t = \Phi(\mathbf{s}_t, \mathbf{b}^{ex}_t) + \mathbf{K}_c(\mathbf{q}^{ex}_t - \Psi(\mathbf{X}^p_t)) \tag{10}$$

Here, $\Phi(\cdot)$ generates nominal joint commands ($\mathbf{q}^{ex}_t$) and torques ($\boldsymbol{\tau}^{ex}_t$), while $\Psi(\cdot)$ provides physics-informed state predictions $\hat{\mathbf{q}}_{t+1}$. The gain matrix $\mathbf{K}_c$ transforms anticipated state discrepancies into compensatory torques, effectively approximating the required disturbance compensation $\Delta\boldsymbol{\tau}_{t+1}$.

During deployment, the values of $\mathbf{K}_c$ are updated in order to minimize the forward looking tracking errors $\Delta\mathbf{q}_{t+1} = \mathbf{q}^{ex}_t - \hat{\mathbf{q}}_{t+1}$ through adaptive control techniques. A bounded-gain-forgetting (BGF) composite adaptive control law [39], [66], [67], [72] is developed:

$$\dot{\mathbf{K}}_c = -\Gamma(t)\Delta\mathbf{q}_{t+1}(\boldsymbol{\epsilon}^\top_{\mathbf{q}} + \kappa\boldsymbol{\epsilon}^\top_\tau) \tag{11}$$

where $\boldsymbol{\epsilon}_{\mathbf{q}} = \dot{\mathbf{q}}_t - \alpha(\mathbf{q}^{ex}_t - \mathbf{q}_t)$ and $\boldsymbol{\epsilon}^\top_\tau = \boldsymbol{\tau}^{ex}_t - \boldsymbol{\tau}_t$ represent the joint position and torque tracking errors, respectively, and $\kappa$ controls the contribution of $\boldsymbol{\epsilon}^\top_\tau$. $\Gamma(t)$ represents a time dependent positive-definite update matrix which evolves according to [72]:

$$\frac{d}{dt}\Gamma(t) = \lambda(t)\Gamma(t) - \Gamma(t)\Delta\mathbf{q}_{t+1}\Delta\mathbf{q}^\top_{t+1}\Gamma(t) \tag{12}$$

with $\lambda(t)$ defined as:

$$\lambda(t) = \lambda_0\left(1 - \frac{\|\Gamma(t)\|}{k_0}\right) \tag{13}$$

$\lambda_0$ and $k_0$ are constants defining the upper bound of $\lambda(t)$ and $\|\Gamma(t)\|$ respectively, ensuring bounded gains. Prior work [66], [67] shows that BGF adaptive update law achieves global asymptotic stability with $\boldsymbol{\epsilon}_{\mathbf{q}}$ and $\boldsymbol{\epsilon}^\top_\tau$ asymptotically converging to zero.

**Algorithm 1:** FLAP Online Execution

**Input:** $\Psi(\cdot), \mathbf{s}_t, \mathbf{b}^{ex}_t, \mathbf{X}^p_t, \kappa, k_0, \lambda_0$
**Output:** $\mathbf{q}^{ex}_t, \boldsymbol{\tau}^{ex}_t$

**while** ***running*** **do**
  $\mathbf{q}^{ex}_t, \boldsymbol{\tau}^{ex}_t \leftarrow \Phi(\mathbf{s}_t, \mathbf{b}^{ex}_t)$, $\hat{\mathbf{q}}_{t+1} \leftarrow \Psi(\mathbf{X}^p_t)$
  **repeat**
    get feedback $\mathbf{q}_t, \dot{\mathbf{q}}_t, \boldsymbol{\tau}_t$
    update $\boldsymbol{\epsilon}_{\mathbf{q}}$ and $\boldsymbol{\epsilon}^\top_\tau$, update $\lambda(t)$ with Eq. 13
    update $\Gamma(t)$ with Eq. 12, update $\mathbf{K}_c$ with Eq. 11
    $\Delta\boldsymbol{\tau}_{t+1} \leftarrow \mathbf{K}_c(\mathbf{q}^{ex}_t - \hat{\mathbf{q}}_{t+1})$
    **return** $\boldsymbol{\tau}^{ex}_t + \Delta\boldsymbol{\tau}_{t+1}$
  **until** *next* $\mathbf{q}^{ex}_t$,$\boldsymbol{\tau}^{ex}_t$, *and* $\hat{\mathbf{q}}_{t+1}$
  update $\mathbf{s}$, $\mathbf{b}^{ex}$, $\mathbf{X}^p$

Algorithm 1 implements FLAP through two parallel processes running at different frequencies: an outer loop (0.5 kHz) generating baseline commands via $\Phi(\cdot)$ and $\Psi(\cdot)$, and an inner loop (1 kHz) executing the BGF adaptation (Eqs. 10-13). The outer loop computes $\mathbf{q}^{ex}_t$, $\boldsymbol{\tau}^{ex}_t$, and $\hat{\mathbf{q}}_{t+1}$ while updating the models inputs $\mathbf{s}_t, \mathbf{b}^{ex}_t, \mathbf{X}^p_t$, while the inner loop applies the BGF update law using the predicted values to produce $\Delta\boldsymbol{\tau}_{t+1}$. This multi-rate architecture enables quickly generating forward looking joint torque compensation values for rapidly adapting $\Phi(\cdot)$ to dynamic payloads.

TABLE I
PAYLOAD $\mathbf{b}^{ex}$ LIMITS DURING TRAINING AND EVALUATION

| **Training** | | **Evaluation** | |
|---|---|---|---|
| Payload | $\mathbf{b}^{ex}$ Limits | Payload | $\mathbf{b}^{ex}$ Limits |
| $<$ 5kg | $\pm[1.0, 0.8, 2.0]$ | $<$ 5kg | $\pm[1.0, 0.8, 2.0]$ |
| $\geq$ 5kg | $\pm[0.5, 0.4, 1.0]$ | $\geq$ 5kg | $\pm[1.0, 0.8, 2.0]$ |

## IV. EXPERIMENTS

### A. Experimental Settings

We experimentally validate FLAP through simulated and physical robot experiments using the Unitree Go1 quadruped robot [73]. $\Phi(\cdot)$ is implemented using the Whole Body Impulse Control (WBIC) method [4] and $\Psi(\cdot)$ is implemented as an MLP with the architecture shown in Fig. 2.

FLAP's training data consists of 2.88 million timesteps (96 minutes) collected in MuJoCo simulations [74], encompassing six payload masses (0, 1.0, 2.5, 5.0, 7.5, and 10 kg) and diverse locomotion behaviors (rotations, forward/backward,

TABLE II

RMSE QUANTITATIVE RESULTS OF FLAP AND COMPARISON METHODS FOR SUCCESSFUL TRAILS ACROSS MULTIPLE PAYLOAD CONDITIONS AND MOTIONS IN SIMULATION. BEST PERFORMANCE IS **BOLDED** AND SECOND BEST IS UNDERLINED.

| **Condition** | **Failure Rate** (/40) | | | | **Torso-Position** ($m$) | | | | **Torso-Orientation** ($rad$) | | | | **Joint-Position** ($rad$) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | WBIC | REACT | F-NP | FLAP | WBIC | REACT | F-NP | FLAP | WBIC | REACT | F-NP | FLAP | WBIC | REACT | F-NP | FLAP |
| **10kg** | 40 | 20 | 20 | **5** | - | 0.0111 | **0.0108** | 0.0110 | - | 0.5391 | **0.3528** | 0.3691 | - | 0.0957 | **0.0793** | 0.0806 |
| **7kg** | 27 | 5 | 3 | **0** | 0.0241 | 0.0108 | 0.0104 | **0.0097** | 0.9887 | 0.4691 | 0.3945 | **0.3456** | 0.2732 | 0.0874 | 0.0781 | **0.0692** |
| **4kg** | 0 | 0 | 0 | 0 | 0.0155 | 0.0094 | 0.0092 | **0.0088** | 0.6108 | 0.4241 | 0.3887 | **0.3347** | 0.1574 | 0.0757 | 0.0707 | **0.0629** |
| **Varied** | 38 | 14 | 9 | **1** | 0.0212 | 0.0110 | 0.0107 | **0.0097** | 1.1136 | 0.5262 | 0.4409 | **0.3447** | 0.2878 | 0.0969 | 0.0852 | **0.0705** |
| **Total Avg.** | 105 | 39 | 32 | **6** | 0.0181 | 0.0104 | 0.0102 | **0.0098** | 0.7399 | 0.4802 | 0.3985 | **0.3481** | 0.1967 | 0.0874 | 0.0779 | **0.0708** |
| **Avg. % Im.** | - | 62.86 | 69.52 | **94.29** | - | 42.54 | 43.65 | **45.86** | - | 35.10 | 46.14 | **52.95** | - | 55.57 | 60.40 | **64.01** |

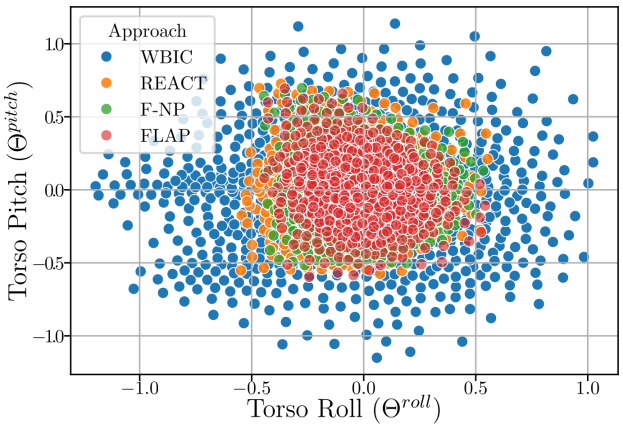


Fig. 3. Scatter plot of gravity projections through torso roll ($\Theta^{roll}$) and pitch ($\Theta^{pitch}$) capturing COM-shift experienced by each approach transporting a 4kg payload.

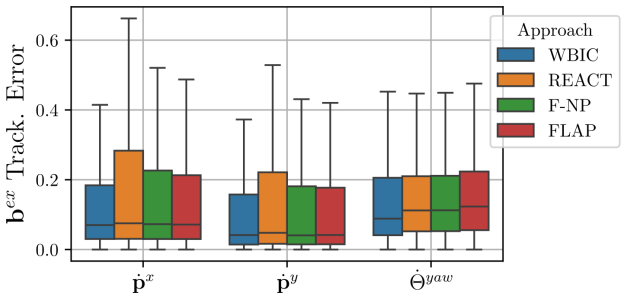


Fig. 4. Tracking error between expected $\mathbf{b}^{ex}$ and actual torso velocity behaviors ($\dot{\mathbf{p}}^x$, $\dot{\mathbf{p}}^y$, $\dot{\Theta}^{yaw}$).

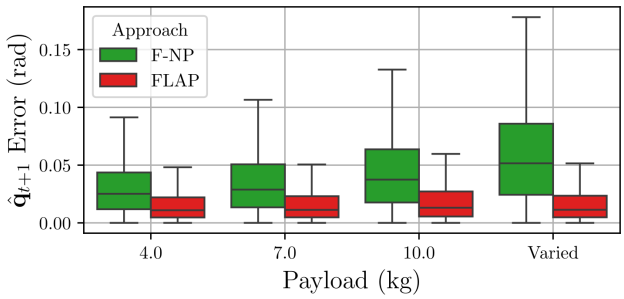


Fig. 5. FLAP and F-NP $\hat{\mathbf{q}}_{t+1}$ prediction errors vs. payload condition.

and lateral movements) on flat terrain. To simulate the effects of a dynamic payload, we applied random torques (resampled at 10 Hz) scaled by payload mass to the robot's torso, creating ground reaction force variations ($\Delta\mathbf{f}^r$) characteristic of the shifting mass distributions of dynamic payloads. During training data collection, we imposed velocity limits on $\mathbf{b}^{ex}$ for payloads $\geq$ 5.0 kg due to $\Phi(\cdot)$'s uncompensated payload capacity; these limits were removed during evaluation (Table I). Training and simulation used an 8-core i9 workstation (64GB RAM), while real-time deployment leveraged an Nvidia Jetson Orin Nano [75] installed on the robot. Complete implementation, training, and evaluation details are available in the supplementary material.

We benchmark FLAP against three approaches: (1) non-adaptive $\Phi(\cdot)$ (**WBIC**) [4], (2) reactive compensation (**REACT**, Eq. 4), and (3) a physics agnostic variant (**F-NP**) using an identical architecture to FLAP but without PINN loss terms $\mathcal{L}^p_{wb}$ (Eq. 8) and $\mathcal{L}^p_{\dot{\mathbf{q}}_{t+1}}$ (Eq. 7). All adaptive methods share the same $\Phi(\cdot)$ and BGF update law. Performance is quantified through failure rates, Root Mean Squared Error (RMSE) of expected and actual torso pose/orientation and leg positions, expected torso velocity tracking accuracy, and percentage improvements over the WBIC baseline.

### B. Results on Simulated Evaluation

We evaluated FLAP and baseline methods in MuJoCo simulations across four payload configurations (4kg, 7kg, 10kg, and 2-10kg variable mass) during four distinct locomotion behaviors (forward/backward, lateral motion, turning, turning while moving) on flat terrain. Performance metrics were computed by averaging results from ten independent trials per condition, with final values aggregated by payload category for successful trials. Each successful trial comprised 23,493 timesteps (23.493 seconds) of continuous operation sampled at 1 kHz.

*1) Tracking Performance:* Table II presents pose tracking performance and failure rates. The forward looking adaptation methods demonstrated consistent superiority over non-adaptive and reactive baselines, with FLAP achieving a 94.29% lower failure rate than WBIC. Although F-NP showed marginally better pose tracking for the 10kg payload, FLAP's significantly lower failure rate (12.5% vs 50%) suggests this difference arises from stochastic payload dynamics rather than controller performance. Fig. 3 further validates FLAP's torso stability advantages, showing tighter clustering of gravity vector projections through the torso's orientation (indicating COM shifts through roll/pitch) during 4kg payload transport compared to other methods. These results collectively demonstrate FLAP's advantages for dynamic payload transportation.

Fig. 4 summarizes expected velocity command tracking across all conditions. FLAP achieves improved linear velocity tracking compared to adaptive baselines, but marginally worse yaw performance. Notably, all adaptive methods underperform

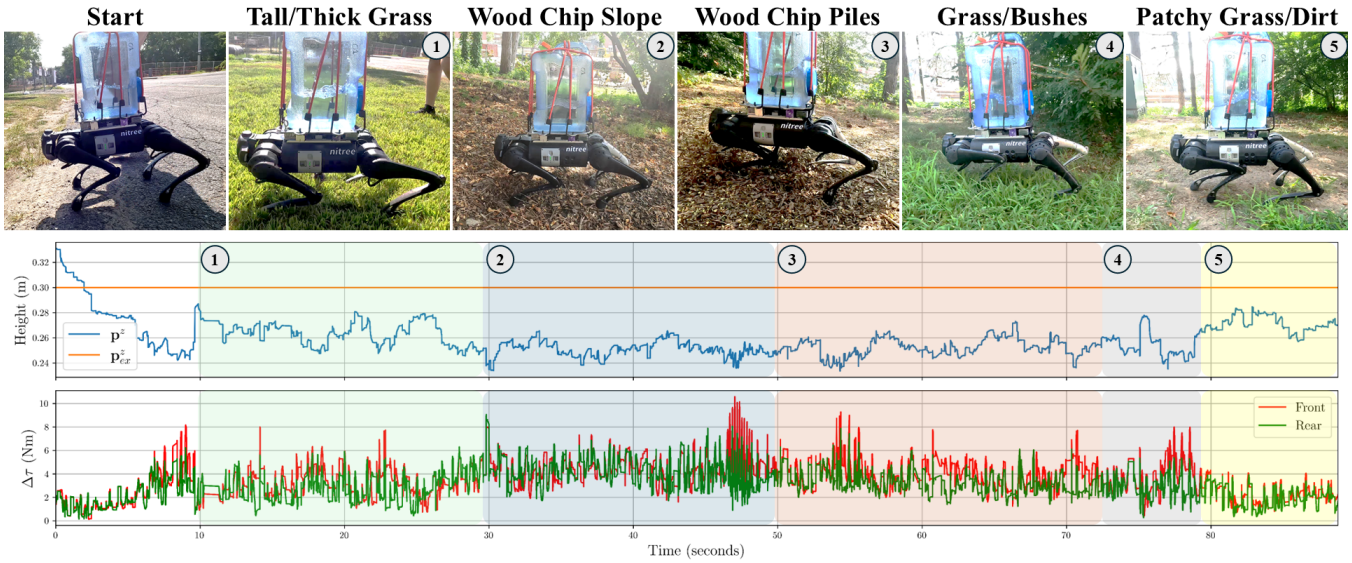


Fig. 6. Torso height tracking performance and the L2 norm of forward looking joint torque compensations (split between front and rear legs) of FLAP while transporting a 6 liter liquid water payload in real-world unstructured terrains.

the non-adaptive WBIC. This is because while compensating for payload disturbances improves torso stability/posture, it also induces slight velocity reductions during corrective phases. Proactive methods narrow the performance gap with WBIC in linear velocity tracking because their anticipatory adjustments require less severe corrections than reactive approaches. However, the enhanced torso stability during turns comes at the cost of reduced turning rates. These results demonstrate a minor trade-off between pose and velocity tracking performance, especially during turning maneuvers.

*2) Forward Looking Prediction Performance:* Fig. 5 compares the $\hat{\mathbf{q}}_{t+1}$ prediction accuracy of FLAP and F-NP under payload and velocity conditions ($\mathbf{b}^{ex}$) beyond the training distribution (where $\mathbf{b}^{ex}$ limits were relaxed per Table I). While F-NP's performance degrades with increasing payload and $\mathbf{b}^{ex}$-induced distribution shift, FLAP maintains stable prediction accuracy through its physics informed loss functions. This demonstrates that PINN-based enforcement of physical consistency enables reliable generalization to novel system states, supporting the advantage of physics informed training.

### C. Real-World Liquid Payload Adaptation Demonstration

We validated FLAP 's robustness to real-world conditions through deployment on a physical robot carrying a 6-liter liquid payload across a varied outdoor course featuring five distinct terrains not encountered during training (Fig. 6). Transporting a liquid payload in the real-world requires not only counteracting the impact of dynamic payload but also the ground reaction force variations introduced from unfamiliar terrains. The robot began on flat pavement before progressing to an uneven lawn where its feet sank into thick grass. It then encountered a walking path with a $13.67°$ incline covered with loose wood chips,. The incline disrupted expected foothold placements and caused the robot pitching forwards. This necessitated substantial compensatory joint torques ($\Delta\tau$), particularly in the front legs. Subsequent sections included large wood chip piles that induced stumbling, tall grass with overhanging bushes that interfered with the payload, and finally patchy mixed terrain. Over the entire 65.45-meter course, completed in 88.79 seconds, FLAP successfully adapted to both the dynamic liquid payload and unpredictable terrain-induced disturbances. In contrast, the baseline controller $\Phi(\cdot)$ alone failed at the first terrain transition, underscoring the limitations of non-adaptive methods in complex real-world conditions. Complete videos of this and more outdoor experiments can be found on the project's website.

## V. Conclusion

We presented FLAP, a physics-informed learning framework for dynamic payload adaptation in quadrupedal locomotion. Through novel PINN-based dynamics modeling and composite adaptive control, FLAP achieves superior performance compared to both reactive and proactive baselines. FLAP achieves a 94.29% lower failure rates than non-adaptive controller, with a 24.77% improvement over the next best approach, and effective generalization to novel system states and real-world terrains through physically consistent representations of the robot's dynamics. While revealing an inherent stability-velocity tradeoff during turning maneuvers, our results establish that embedding known dynamics into the learning process enables robust performance in real-world dynamic payload transportation scenarios. Future work will explore extending this framework to more complex terrain and payload configurations.

## References

[1] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[2] 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," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[3] 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.

[4] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *arXiv preprint arXiv:1909.06586*, 2019.

[5] 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.

[6] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *Robotics: Science and Systems*, 2021.

[7] M. Seo, R. Gupta, Y. Zhu, A. Skoutnev, L. Sentis, and Y. Zhu, "Learning to walk by steering: Perceptive quadrupedal locomotion in dynamic environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[8] D. Hoeller, N. Rudin, D. Sako, and M. Hutter, "Anymal parkour: Learning agile navigation for quadrupedal robots," *Science Robotics*, vol. 9, no. 88, p. eadi7566, 2024.

[9] Y. Fan, Z. Pei, C. Wang, M. Li, Z. Tang, and Q. Liu, "A review of quadruped robots: Structure, control, and autonomous motion," *Advanced Intelligent Systems*, vol. 6, no. 6, p. 2300783, 2024.

[10] C. Gehring, P. Fankhauser, L. Isler, R. Diethelm, S. Bachmann, M. Potz, L. Gerstenberg, and M. Hutter, "Anymal in the field: Solving industrial inspection of an offshore hvdc platform with a quadrupedal robot," in *Field and Service Robotics*. Springer, 2021, p. 247.

[11] S. Halder, K. Afsari, J. Serdakowski, S. DeVito, M. Ensafi, and W. Thabet, "Real-time and remote construction progress monitoring with a quadruped robot using augmented reality," *Buildings*, vol. 12, no. 11, p. 2027, 2022.

[12] K. Afsari, S. Halder, R. King, W. Thabet, J. Serdakowski, S. DeVito, M. Ensafi, and J. Lopez, "Identification of indicators for effectiveness evaluation of four-legged robots in automated construction progress monitoring," in *Construction Research Congress*, 2022.

[13] S. Halder, K. Afsari, E. Chiou, R. Patrick, and K. A. Hamed, "Construction inspection & monitoring with quadruped robots in future human-robot teaming: A preliminary study," *Journal of Building Engineering*, vol. 65, p. 105814, 2023.

[14] R. Siegwart, M. Hutter, P. Oettershagen, M. Burri, I. Gilitschenski, E. Galceran, and J. Nieto, "Legged and flying robots for disaster response," in *World Engineering Conference and Convention (WECC)*, 2015.

[15] T. Klamt, D. Rodriguez, M. Schwarz, C. Lenz, D. Pavlichenko, D. Droeschel, and S. Behnke, "Supervised autonomous locomotion and manipulation for disaster response with a centaur-like robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[16] C. D. Bellicoso, M. Bjelonic, L. Wellhausen, K. Holtmann, F. Günther, M. Tranzatto, P. Fankhauser, and M. Hutter, "Advances in real-world applications for legged robots," *Journal of Field Robotics*, vol. 35, no. 8, pp. 1311–1326, 2018.

[17] J. Delmerico, S. Mintchev, A. Giusti, B. Gromov, K. Melo, T. Horvat, C. Cadena, M. Hutter, A. Ijspeert, D. Floreano *et al.*, "The current state and future outlook of rescue robotics," *Journal of Field Robotics*, vol. 36, no. 7, pp. 1171–1191, 2019.

[18] G. Tournois, M. Focchi, A. Del Prete, R. Orsolino, D. G. Caldwell, and C. Semini, "Online payload identification for quadruped robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[19] M. Sombolestan, Y. Chen, and Q. Nguyen, "Adaptive force-based control for legged robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[20] M. V. Minniti, R. Grandia, F. Farshidian, and M. Hutter, "Adaptive clf-mpc with application to quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 565–572, 2021.

[21] M. Sombolestan and Q. Nguyen, "Adaptive-force-based control of dynamic legged locomotion over uneven terrain," *IEEE Transactions on Robotics*, vol. 40, pp. 2462–2477, 2024.

[22] X. B. Peng, E. Coumans, T. Zhang, T.-W. E. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," in *Robotics: Science and Systems*, 2020.

[23] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science Robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[24] Z. Zhuang, Z. Fu, J. Wang, C. G. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao, "Robot parkour learning," in *Conference on Robot Learning*, 2023.

[25] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[26] Y. Yang, K. Caluwaerts, A. Iscen, T. Zhang, J. Tan, and V. Sindhwani, "Data efficient reinforcement learning for legged robots," in *Conference on Robot Learning*, 2020.

[27] X. Song, Y. Yang, K. Choromanski, K. Caluwaerts, W. Gao, C. Finn, and J. Tan, "Rapidly adaptable legged robots via evolutionary meta-learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[28] L. Smith, J. C. Kew, X. B. Peng, S. Ha, J. Tan, and S. Levine, "Legged robots that keep on learning: Fine-tuning locomotion policies in the real world," in *IEEE/RSJ International Conference on Robotics and Automation (ICRA)*, 2022.

[29] L. Smith, J. C. Kew, T. Li, L. Luu, X. B. Peng, S. Ha, J. Tan, and S. Levine, "Learning and adapting agile locomotion skills by transferring experience," *arXiv preprint arXiv:2304.09834*, 2023.

[30] L. Smith, Y. Cao, and S. Levine, "Grow your limits: Continuous improvement with real-world rl for robotic locomotion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[31] L. Amanzadeh, T. Chunawala, R. T. Fawcett, A. Leonessa, and K. A. Hamed, "Predictive control with indirect adaptive laws for payload transportation by quadrupedal robots," *IEEE Robotics and Automation Letters*, 2024.

[32] C. Cao and N. Hovakimyan, "Design and analysis of a novel l1 adaptive control architecture with guaranteed transient performance," *IEEE Transactions on Automatic Control*, vol. 53, no. 2, pp. 586–591, 2008.

[33] R. J. Full and D. E. Koditschek, "Templates and anchors: neuromechanical hypotheses of legged locomotion on land," *Journal of experimental biology*, vol. 202, no. 23, pp. 3325–3332, 1999.

[34] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Conference on Robot Learning*, 2020.

[35] I. M. A. Nahrendra, B. Yu, and H. Myung, "Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning," in *2023 IEEE International Conference on Robotics and Automation, ICRA 2023*. Institute of Electrical and Electronics Engineers Inc., 2023, pp. 5078–5084.

[36] Y. Song, S. bae Kim, and D. Scaramuzza, "Learning quadruped locomotion using differentiable simulation," in *8th Annual Conference on Robot Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=XopATjibyz

[37] Y. Chen and Q. Nguyen, "Learning agile locomotion and adaptive behaviors via rl-augmented mpc," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[38] X. Zeng, H. Zhang, L. Yue, Z. Song, L. Zhang, and Y.-H. Liu, "Adaptive model predictive control with data-driven error model for quadrupedal locomotion," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5731–5737.

[39] S. Lyu, X. Lang, H. Zhao, H. Zhang, P. Ding, and D. Wang, "Rl2ac: Reinforcement learning-based rapid online adaptive control for legged robot robust locomotion," *Robotics: Science and Systems (RSS)*, 2024.

[40] V. K. Kurva and S. Kolathaya, "Mule: Multi-terrain and unknown load adaptation for effective quadrupedal locomotion," *arXiv preprint arXiv:2505.00488*, 2025.

[41] Y. Zhang, B. Nie, Z. Cao, Y. Fu, and Y. Gao, "Disturbance-aware adaptive compensation in hybrid force-position locomotion policy for legged robots," *arXiv preprint arXiv:2506.00472*, 2025.

[42] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[43] Z. Mao, A. D. Jagtap, and G. E. Karniadakis, "Physics-informed neural networks for high-speed flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 360, p. 112789, 2020.

[44] M. Mahmoudabadbozchelou, G. E. Karniadakis, and S. Jamali, "nn-pinns: Non-newtonian physics-informed neural networks for complex fluid modeling," *Soft Matter*, vol. 18, no. 1, pp. 172–185, 2022.

[45] C. Zhao, F. Zhang, W. Lou, X. Wang, and J. Yang, "A comprehensive review of advances in physics-informed neural networks and their

applications in complex fluid dynamics," *Physics of Fluids*, vol. 36, no. 10, 2024.

[46] S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks for heat transfer problems," *Journal of Heat Transfer*, vol. 143, no. 6, p. 060801, 2021.

[47] D. Jalili, S. Jang, M. Jadidi, G. Giustini, A. Keshmiri, and Y. Mahmoudi, "Physics-informed neural networks for heat transfer prediction in two-phase flows," *International Journal of Heat and Mass Transfer*, vol. 221, p. 125089, 2024.

[48] J. Wang, Y. Mo, B. Izzuddin, and C.-W. Kim, "Exact dirichlet boundary physics-informed neural network epinn for solid mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 414, p. 116184, 2023.

[49] J. Bai, T. Rabczuk, A. Gupta, L. Alzubaidi, and Y. Gu, "A physics-informed neural network technique based on a modified loss function for computational 2d and 3d solid mechanics," *Computational Mechanics*, vol. 71, no. 3, pp. 543–562, 2023.

[50] H. Hu, L. Qi, and X. Chao, "Physics-informed neural networks (pinn) for computational solid mechanics: Numerical frameworks and applications," *Thin-Walled Structures*, vol. 205, p. 112495, 2024.

[51] L. Wang, G. Liu, G. Wang, and K. Zhang, "M-pinn: A mesh-based physics-informed neural network for linear elastic problems in solid mechanics," *International journal for numerical methods in engineering*, vol. 125, no. 9, p. e7444, 2024.

[52] J. Liu, P. Borja, and C. Della Santina, "Physics-informed neural networks to model and control robots: A theoretical and experimental investigation," *Advanced Intelligent Systems*, vol. 6, no. 5, p. 2300385, 2024.

[53] J. Nicodemus, J. Kneifl, J. Fehr, and B. Unger, "Physics-informed neural networks-based model predictive control for multi-link manipulators," *IFAC-PapersOnLine*, vol. 55, no. 20, pp. 331–336, 2022.

[54] W. Deng, F. Ardiani, K. T. Nguyen, M. Benoussaad, and K. Medjaher, "Physics informed machine learning model for inverse dynamics in robotic manipulators," *Applied Soft Computing*, vol. 163, p. 111877, 2024.

[55] S. Sanyal and K. Roy, "Ramp-net: A robust adaptive mpc for quadrotors via physics-informed neural network," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1019–1025.

[56] W. Gu, S. Primatesta, and A. Rizzo, "Physics-informed neural network for quadrotor dynamical modeling," *Robotics and Autonomous Systems*, vol. 171, p. 104569, 2024.

[57] M. Lahariya, C. Innes, C. Develder, and S. Ramamoorthy, "Learning physics-informed simulation models for soft robotic manipulation: A case study with dielectric elastomer actuators," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 11 031–11 038.

[58] W. Sun, N. Akashi, Y. Kuniyoshi, and K. Nakajima, "Physics-informed recurrent neural networks for soft pneumatic actuators," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6862–6869, 2022.

[59] X. Wang, J. J. Dabrowski, J. Pinskier, L. Liow, V. Viswanathan, R. Scalzo, and D. Howard, "Pinn-ray: A physics-informed neural network to model soft robotic fin ray fingers," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 247–254.

[60] T.-L. Habich, A. Mohammad, S. F. Ehlers, M. Bensch, T. Seel, and M. Schappler, "Generalizable and fast surrogates: Model predictive control of articulated soft robots using physics-informed neural networks," *arXiv preprint arXiv:2502.01916*, 2025.

[61] X. Yang, Y. Du, L. Li, Z. Zhou, and X. Zhang, "Physics-informed neural network for model prediction and dynamics parameter identification of collaborative robot joints," *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 8462–8469, 2023.

[62] Y. Liu, Y. Bao, P. Cheng, D. Shen, G. Chen, and H. Xu, "Enhanced robot state estimation using physics-informed neural networks and multimodal proprioceptive data," in *Sensors and Systems for Space Applications XVII*, vol. 13062. SPIE, 2024, pp. 144–160.

[63] I. Sorrentino, G. Romualdi, L. Moretti, S. Traversaro, and D. Pucci, "Physics-informed neural networks with unscented kalman filter for sensorless joint torque estimation in humanoid robots," *IEEE Robotics and Automation Letters*, 2025.

[64] R. Ni and A. H. Qureshi, "Physics-informed neural motion planning on constraint manifolds," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 12 179–12 185.

[65] Y. Liu, R. Ni, and A. H. Qureshi, "Physics-informed neural mapping and motion planning in unknown environments," *IEEE Transactions on Robotics*, 2025.

[66] J.-J. E. Slotine and W. Li, "Composite adaptive control of robot manipulators," *Automatica*, vol. 25, no. 4, pp. 509–519, 1989.

[67] Y. Pan and H. Yu, "Composite learning robot control with guaranteed parameter convergence," *Automatica*, vol. 89, pp. 398–406, 2018.

[68] Y. Song, S. bae Kim, and D. Scaramuzza, "Learning quadruped locomotion using differentiable simulation," in *Conference on Robot Learning*. PMLR, 2025, pp. 258–271.

[69] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," 2018. [Online]. Available: http://github.com/jax-ml/jax

[70] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, 2019.

[71] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[72] J. Zhang, W. Liu, L. Gao, L. Li, and Z. Li, "The master adaptive impedance control and slave adaptive neural network control in underwater manipulator uncertainty teleoperation," *Ocean Engineering*, vol. 165, pp. 465–479, 2018.

[73] [Online]. Available: https://www.unitree.com/go1

[74] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[75] [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/nano-super-developer-kit/