Robotics: Science and Systems 2025
Los Angeles, CA, USA, June 21–25, 2025

SATA: Safe and Adaptive Torque-Based Locomotion Policies Inspired by Animal Learning

Peizhuo LI[1*], Hongyi LI[1*], Ge SUN[1], Jin CHENG[2], Xinrong YANG[1], Guillaume Bellegarda[3], Milad Shafiee[3], Yuhong CAO[1†], Auke Ijspeert[3], Guillaume SARTORETTI[1]
[*]Equal contribution, [†]Corresponding author,
[1]MARMot Lab, National University of Singapore [2]Computational Robotics Lab, ETH Zurich [3]BioRob Lab, EPFL
Video: https://youtu.be/b1cpTq0Rc5w Code:https://github.com/marmotlab/SATA

***Abstract*—Despite recent advances in learning-based controllers for legged robots, deployments in human-centric environments remain limited by safety concerns. Most of these approaches use position-based control, where policies output target joint angles that must be processed by a low-level controller (e.g., PD or impedance controllers) to compute joint torques. Although impressive results have been achieved in controlled real-world scenarios, these methods often struggle with compliance and adaptability when encountering environments or disturbances unseen during training, potentially resulting in extreme or unsafe behaviors. Inspired by how animals achieve smooth and adaptive movements by controlling muscle extension and contraction, torque-based policies offer a promising alternative by enabling precise and direct control of the actuators in torque space. In principle, this approach facilitates more effective interactions with the environment, resulting in safer and more adaptable behaviors. However, challenges such as a highly nonlinear state space and inefficient exploration during training have hindered their broader adoption. To address these limitations, we propose Safe and Adaptive Torque-based locomotion policies inspired by Animal learning(SATA), a bio-inspired framework that mimics key biomechanical principles and adaptive learning mechanisms observed in animal locomotion. Our approach effectively addresses the inherent challenges of learning torque-based policies by significantly improving early-stage exploration, leading to high-performance final policies. Remarkably, our method achieves zero-shot sim-to-real transfer, eliminating the need for additional fine-tuning on hardware. Our experimental results indicate that SATA demonstrates remarkable compliance and safety, even in challenging environments such as soft/slippery terrain or narrow passages, and under significant external disturbances (e.g., pushing/pulling/pressing on the robot, or manually moving individual legs). These results highlight its potential for practical deployments in human-centric and safety-critical scenarios.**

I. Introduction

Reinforcement learning (RL) has demonstrated significant potential in the control of legged robots [1, 2]. Compared to conventional control methods, such as Model Predictive Control (MPC), RL-based approaches exhibit remarkable robustness, enabling quadrupedal robots to navigate effectively in complex terrain [3, 4, 5].

Most existing RL-based methods rely on position control [6, 7]. In this approach, neural networks output target joint positions, which are subsequently translated into joint torque commands through a low-level proportional–derivative (PD) controller. Position-based policies are simple and easy to train, as they abstract away the complexities of actuation physics and dynamics. However, this simplicity limits the policy's capacity to explore fine-grained and dynamic behaviors, thereby reducing its adaptability and generalization to unseen challenges in real-world environments. For instance, position-based policies trained on rigid terrains in simulation often struggle to generalize to deformable environments in the real world due to their out-of-distribution nature. As a result, these methods usually rely heavily on terrain randomization, where the agent trains in large amounts of potential environments to maximize the chances that the final policy may handle real-world environments [8, 9]. Furthermore, position-based strategies exhibit notable limitations in compliance. Lacking the ability to directly regulate joint torques, these methods often lead to overreactions to external disturbances. For example, if one of the robot's legs suddenly gets stuck, a position-based policy may aggressively attempt to force the actuator into the commanded position. This can cause the motors to generate excessively high torques, potentially destabilizing or even damaging the robot, or creating safety hazards to its

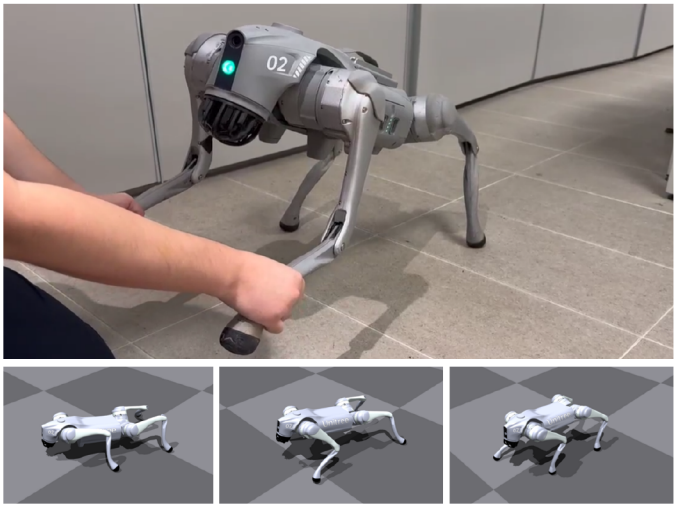

Fig. 1. Inspired by biomechanical principles and the growth mechanisms of animals in nature, we propose a framework that addresses the challenges of torque-based locomotion learning, achieving zero-shot sim-to-real transfer along with exceptional compliance and safety in challenging environments.

surroundings, such as nearby objects or even humans.

In contrast, torque-based learning controllers aim to train a policy network that directly generates torques for all joints, providing enhanced compliance and adaptability. By directly controlling actuation in torque space, this approach enables finer interaction with the environment, leading to more dynamic and robust locomotion. Moreover, torque control allows the robot to explicitly reason about its dynamics, theoretically offering a greater ability to handle unknown environments and abrupt disturbances without relying on predefined low-level controllers [10]. However, torque-based policies come with their own set of challenges, including a high-dimensional action space and greater nonlinear transformations from states to actions. These factors make exploration substantially harder during training, particularly in the early stages. During this phase, the abundance of local optima can obstruct the exploration process, leading to premature convergence and unnatural gait behaviors. Few works have effectively addressed these challenges. DeCAP, proposed by Sood et al. [11], mitigates these issues by leveraging pre-trained position-based policies. However, its increased training complexity limits its broader adoption. Similarly, Chen et al. [12] successfully trained a torque policy by incorporating additional reward terms and action scaling. Yet, this approach remains highly sensitive to hyperparameter tuning and often exhibits low exploration efficiency during the initial stages of training, preventing it from consistently yielding high-performance policies.

To overcome these limitations, we propose enhancements to directly learn torque-based policies by drawing inspiration from biological systems. In animals, smooth motion actuation is achieved through the intrinsic biomechanical properties of muscles, such as the force-velocity relationship described by the Hill model [13], which regulates movement commands and prevents excessive behaviors. Moreover, certain muscle mechanisms provide feedback that can aid decision-making, such as muscle fatigue (Feedback of pain due to prolonged exertion) [14]. Borrowing from these mechanisms, we integrate a simplified biomechanical model into our torque-based learning framework. This model retains the functional characteristics of biological muscles, enabling the robot to perform smoother actions while mitigating the risk of suboptimal convergence. In addition, we incorporate a growth mechanism inspired by the gradual development processes observed when animals learn to locomote. This mechanism dynamically adjusts key robot properties during training, such as torque limits and control frequency, while also modulating the reward terms throughout the process. This significantly enhances early-stage exploration and improves the generalizability of the trained policy.

By addressing the inherent challenges in torque-based policy learning, our approach not only provides a robust and efficient solution for torque-based control but also demonstrates high performance in compliance and adaptability to previously unseen scenarios, such as locomotion on deformable terrains. These results highlight the potential of torque-based controllers to surpass the limitations of position-based methods, enabling safe and robust locomotion in complex environments.

The main contributions of this work are as follows:

- **Stable and Efficient Torque-Based Learning:** We propose a novel framework for learning torque-based locomotion policies with a growth mechanism that gradually unlocks torque limits, control frequency, and reward terms, enhancing sample efficiency and training stability in torque space.
- **Biomechanical Model and Safety:** We implement a simplified biomechanical model for actuators that ensures smooth and safe behaviors, even under disturbances and performs robustly in diverse environments.
- **Generalization:** Our experimental results demonstrate that SATA generalizes effectively to out-of-distribution terrains and commands while exhibiting exceptional compliance during human interactions. These findings underscore the robustness and versatility of our approach.

II. Related Work

Quadruped controllers have greatly benefited from the development of deep reinforcement learning (DRL) in recent years, allowing agents to learn impressive controllers [6, 15, 16, 17] that would otherwise require the design and solving of non-linear optimization problems, which often involves approximations that cannot be neglected in real-world settings [18, 19, 20, 21]. However, the problem of sampling efficiency still hinders its application. Early approaches employed by Peng et al. [22] combined imitation learning with DRL to solve this problem, demonstrating gaits similar to the training set while adapting to different terrain or disturbances. This approach allows for the training of adaptive quadruped controllers in highly dynamic acrobatic tasks, but its performance is limited by the dataset quality and robot deployment remains nontrivial. Addressing these problems, Hwangbo et al. [23] proposed a method to train deployable policies in simulation by introducing an *actuator network* and domain randomization. Their learned policies can yield different gaits or recover from fallen positions. To help the robot travel in challenging terrains, Lee et al. [4] introduced a teacher-student framework for quadrupedal robots, allowing them to traverse complex terrains without any visual feedback. To aid locomotion with environmental information, [5] built upon [4] by integrating additional sensors and employing an attention-based recurrent encoder to fuse proprioceptive and exteroceptive inputs. This resulted in a robust and fast legged motion controller for navigating challenging terrains. While exteroceptive inputs can enable more informed decision-making, a highly compliant and robust base policy—capable of operating effectively without vision—remains crucial for reliable real-world deployment. Moreover, reliance on exteroception introduces additional challenges, such as the sim-to-real gap, where sensor noise, latency, and real-world variations degrade performance.

Learning-based controllers typically use position-based action spaces, where the policy directly outputs position commands for the actuators. These commands are subsequently converted to torque using a low-level (e.g., PD) controller

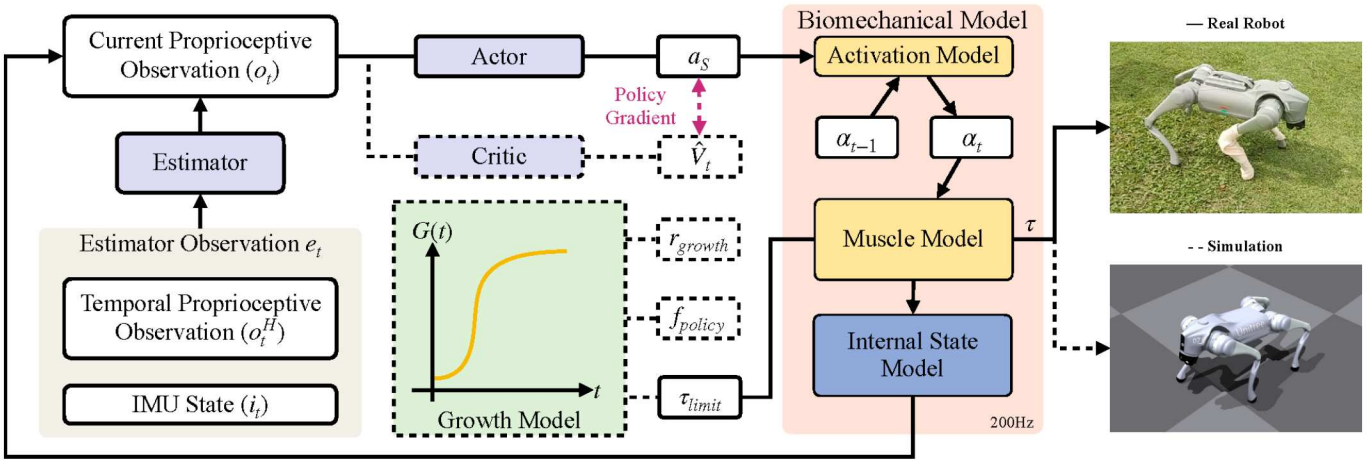

Fig. 2. Overview of our SATA Framework. Dotted lines indicate parts used only during training, while solid lines indicate those used during both training and deployment. Our framework is mainly composed of 1) a Biomechanical Model (Orange) to ensure the generation of smooth, practical actuator commands τ while informing the policy of the current actuator state, and 2) a Growth Model (Green) to help the neural network train a more robust and generalizable policy by gradually adapting rewards r_{growth}, control frequency f_{policy}, and torque limits τ_{limit} during training. Finally, we train a state estimator for real world deployment using simulated IMU data and temporal proprioception observations (Grey), to help condition our policy on the (estimated) current robot velocity.

during training [24, 25, 26]. While such low-level controllers facilitate early-stage exploration for reinforcement learning policies, extensive parameter tuning is often required to ensure successful deployment [27, 28]. Moreover, position control tends to treat the robot as a rigid system, which can result in significant joint or structural stress when operating in uncertain environments (e.g. slipping, collisions), thereby increasing the risk of system damage [29].

In contrast, torque-based policies, where the policy directly outputs motor torques, eliminate the need for tuning low-level controller parameters. They also take advantage of the inherent compliance of torque control to reduce structural stress and impact forces, enhancing the safety of human-robot interactions [30]. There, Chen et al.[12] achieved the first successful sim-to-real transfer of end-to-end torque control for quadrupedal locomotion, enabling RL policies to directly predict joint torques at high frequencies. However, Kim et al.[31] highlighted that torque-based state spaces exhibit significant non-linearity and the controllers need to operate at much higher frequencies, which impairs exploration efficiency during early-stage training.

With the advancement of parallel simulation techniques and high-performance computing [32, 33], massively parallel RL environments have significantly improved sampling efficiency, reducing the training duration to mere minutes [34]. This allows training of more complicated frameworks that can be deployed on-robot easily [35, 36, 37]. To enhance performance, researchers have explored the integration of learning-based approaches with traditional control techniques. For instance, Gangapurwala et al. [38] proposed RL2AC, which combines reinforcement learning policy with an adaptive torque compensator to mitigate external disturbances and model mismatches caused by the sim-to-real gap. Similar studies have also demonstrated that integrating the adaptability of learning-based methods with the robustness of traditional models can significantly improve the locomotion performance of quadruped robots in complex environments [39, 40, 41].

The combination of bio-inspired control and reinforcement learning is another major direction to address the adaptability challenges of legged robots in dynamic and complex environments [42, 43]. This hybrid approach integrates structured prior knowledge from bio-inspired control within reinforcement learning to optimize high-level strategies and key parameters, enabling efficient robot control [44, 45, 46, 47]. For instance, Margolis et al.[9] achieved agile locomotion, including walking, trotting, and pronking, by mimicking natural gait patterns through carefully designed rewards. Similarly, Peng et al. [48] employed imitation learning to teach legged robots agile skills by mimicking real-world animals, while Fu et al. [49] leveraged deep reinforcement learning to minimize energy consumption and achieve emergent gaits. Inspired by Central Pattern Generators (CPGs) observed in animals, Bellegarda et al.[50] combined CPGs with a DRL framework to generate robust and omnidirectional locomotion. Building on this foundation, Sun et al. [51] introduced the learning-based hierarchical control framework, utilizing a spinal policy to adjust CPG frequency and amplitude for rhythmic movement, and a descending modulation policy to adaptively modify rhythmic outputs for precise control on challenging terrains. Similar to this idea, various hierarchical methods have also been proposed for legged controllers [52, 53, 54]. Although these studies have achieved promising results, most of them focus on mimicking animal behaviors or neural structures. Relatively limited research has explored the effects of biomechanical properties and growth processes on the development of locomotion skills in animals, let alone incorporating them into the learning framework for legged robots.

III. Bio-Inspired Neural Architecture

To achieve robust and adaptive locomotion control in legged robots, we propose a bio-inspired neural architecture that em-

ulates key principles of biological systems. This architecture comprises two core modules: the Neural Networks and the Biomechanical Model, as illustrated in Fig. 2. While the Neural Networks generate action signals based on proprioceptive information (see detail in section III-B), the Biomechanical Model is designed to process these action signals, introducing activation dynamics and muscle-like force modulation to ensure smooth and realistic control signals. Additionally, the Biomechanical Model provides internal states feedback to the Neural Networks, providing more temporal information and improved overall performance.

A. Biomechanical Model

Compared to directly using the neural network's output as joint torques, our approach aims to reduce exploration difficulty during training and improve motion continuity. To achieve this, we refine the action signal a_s generated by the neural network using a biomechanical model that employs a biologically inspired two-step process and incorporates a feedback mechanism, including a fatigue mechanism. This fatigue mechanism, inspired by biological systems, dynamically quantifies actuator load and recovery, contributing to more balanced and robust control. The biomechanical model ensures biologically plausible and stable locomotion through three key components.

- **Activation Model**: Converts action signals into intermediate activation signals, incorporating temporal smoothing to reflect the sequential nature of biological systems. This process ensures continuity and prepares signals for precise torque generation.
- **Muscle Model**: Transforms activation signals into joint torques by loosely approximating muscle dynamics. This approach limits torque output to a safe range, preventing abrupt changes that could destabilize the system.
- **Internal State Model**: Tracks the fatigue state of actuators, providing real-time feedback to the neural network. This feedback helps to optimize load distribution, enhancing stability during training and deployment.

By combining these components, the biomechanical model improves exploration efficiency, reduces the risk of local optima, and bridges the sim-to-real transfer gap, making torque commands practical and effective for real-world applications.

1) Activation Model: Output by our policy network, the action signal a_s first passes through the activation model [55]. This model functions similarly to motor neurons [56, 57], transforming the action signal into the corresponding activation signal, α_{current}:

$$\alpha_{\text{current}} = \tanh\left(\frac{a_s \cdot \kappa_{\text{scale}}}{\tau_{\text{limit}}}\right). \tag{1}$$

Here, κ_{scale} represents a scaling factor analogous to the gain of motor neurons in translating neural commands into muscle activations, and τ_{limit} denotes the torque limits of each motor. The resulting activation signal, constrained within the range $(-1, 1)$, emulates the antagonistic coordination of joint control by opposing muscle groups. For example, 1 represents full contraction of one muscle and complete relaxation of its antagonist, resulting in a clockwise torque, while -1 is the exact opposite, enabling precise joint torque modulation.

To ensure smooth and continuous activation dynamics, this model incorporates a temporal update mechanism inspired by the "Hysteresis Effect" [58]. This mechanism accounts for the influence of prior activations on the current state, ensuring stability and natural transitions in movement. The simplified first order temporal evolution of the activation signal α_t is governed by:

$$\alpha_t = (\alpha_{\text{current}} \cdot \gamma) + (\alpha_{t-1} \cdot (1 - \gamma)). \tag{2}$$

In this formulation, γ serves as a smoothing factor. This ensures a smooth temporal evolution of activation signals, which promotes continuity and smoothness in movements.

2) Muscle Model: The activation signal is subsequently passed to the muscle model, where it is transformed into joint torques. There, we developed a dynamics model loosely inspired by the classical Hill muscle model to mitigate potential extreme behaviors [59]. Specifically, we focused on modeling and optimizing the force-velocity relationship in the Hill model [60, 61]. When the activation signal aligns with the joint motion direction, the torque output decreases as the joint velocity increases, effectively suppressing rapid and extreme torque signals. Conversely, when the joint velocity opposes the activation signal, the model generates greater torque to drive the joint towards the desired motion velocity. We believe this force-velocity relationship enhances the activation network's sensitivity to dynamic motion states, reducing instability risks during training and improving exploration efficiency.

Specifically, the joint torque τ is computed as:

$$\tau = \tau_{\text{limit}} \cdot \alpha_t \cdot \left(1 - \text{sign}(\alpha_t) \cdot \frac{\dot{q}}{\dot{q}_{\text{limit}}}\right), \tag{3}$$

where $\dot{q}$ represents the joint velocity, and $\dot{q}_{\text{limit}}$ is the maximum velocity of each joint. This model integrates the activation signal with the dynamic characteristics of joint motion, enabling accurate modeling of joint behavior and ultimately improving the stability and efficiency of the overall control system.

3) Internal State Model: Our internal state model does not directly participate in torque calculation but serves as a feedback mechanism to provide the robot with more information. In this module, we simulate the fatigue mechanism observed in animals to construct a dynamic state indicator that quantifies the accumulation and recovery of fatigue during the robot's operation [62, 63]. The fatigue indicator is directly influenced by the joint torque intensity and evolves dynamically over time, following the equation:

$$\zeta_t = (\zeta_{t-1} + |\tau| \cdot dt) \cdot \beta, \tag{4}$$

where ζ represents the fatigue state, dt the time step, and β the recovery factor, describing the rate at which fatigue dissipates over time.

Through this mechanism, the robot can continuously update its fatigue state, providing the neural network with a dynamic

feedback signal. This fatigue feedback effectively optimizes the control strategy, preventing certain actuators from operating under prolonged high loads, which could otherwise lead to excessive wear or reduced performance. Furthermore, it enables the robot to distribute motion loads more efficiently on each leg during task execution, reducing the occurrence of suboptimal behaviors, such as over-reliance on three-legged or two-legged motion patterns during exploration.

B. Neural Networks

Our SATA framework comprises three sub-networks: actor, critic, and estimator. The actor and critic are optimized using the Proximal Policy Optimization (PPO) algorithm, while the estimator is trained separately, using supervised learning.

1) Observation and Action Space for Actor: The proprioceptive observation vector $\boldsymbol{o_t} \in \mathbb{R}^{60}$ serves as the input to the actor network, encapsulating comprehensive information about the robot's state.

The observation vector is structured as:

$$o_t = [v_t, w_t, g_t, q, \dot{q}, v_{\text{cmd}}, \tau, \zeta]^T,$$

where $v_t \in \mathbb{R}^3$ represents the linear velocity of the robot base. During training, v_t is directly sourced from the simulator, while during deployment, it is estimated by the estimator network. The estimator uses historical proprioceptive and inertial data, structured as:

$$e_t = [[o^H_{t-10}, i_{t-10}], ..., [o^H_t, i_t]]^T.$$

Here, $o^H_t = [q, \dot{q}]$ includes the joint angles and velocities, and $i_t = [\mathbf{a}_t, \boldsymbol{\omega}_t, \mathbf{g}_t]$ represents the IMU data—featuring linear acceleration $\mathbf{a}_t \in \mathbb{R}^3$, angular velocity $\boldsymbol{\omega}_t \in \mathbb{R}^3$, and gravity direction $\mathbf{g}_t \in \mathbb{R}^3$. This design enables robust velocity estimation during deployment.

Other components of o_t include $w_t \in \mathbb{R}^3$, the angular velocity of the robot base, and $g_t \in \mathbb{R}^3$, the gravity direction vector in the body frame. These quantities aid in orientation estimation and maintaining balance. Additionally, $v_{\text{cmd}} \in \mathbb{R}^3$ specifies the desired linear and angular velocities, while $q \in \mathbb{R}^{12}$, $\dot{q} \in \mathbb{R}^{12}$, $\tau \in \mathbb{R}^{12}$, and $\zeta \in \mathbb{R}^{12}$ represent the joint states, joint velocities, joint torques, and fatigue state, as described earlier.

Based on o_t, the actor policy generates the action $a_s \in \mathbb{R}^{12}$, which represents the desired joint torques. These torques are further refined by the biomechanical model to ensure smooth and stable control.

2) Reward Design: In this work, we adopt a relatively simple reward structure, made up of 9 terms designed to effectively encourage natural and robust locomotion. These rewards are categorized into two types: locomotion objectives and auxiliary posture maintenance rewards. The details are summarized in Table I.

There, $\phi(x) = e^{-4 \cdot |x|}$ represents a Gaussian-shaped function used to penalize deviations between actual and commanded values. h_b and h_t denote the robot's base height and target base height above the ground, respectively. $q_{\min}$ and $q_{\max}$ define the lower and upper limits of each joint, while $\ddot{q}$ represents the joint acceleration.

TABLE I
REWARD COMPONENTS AND WEIGHTS ($dt = 0.005$)

Reward Terms	Equation	Weight
Locomotion Objectives		
$r_{\text{tracking},x}$	$\phi\left(v_x - v_x^{\text{cmd}}\right)$	$10dt$
$r_{\text{tracking},y}$	$\phi\left(v_y - v_y^{\text{cmd}}\right)$	$5dt$
$r_{\text{tracking,yaw}}$	$\phi\left(\omega_{\text{yaw}} - \omega_{\text{yaw}}^{\text{cmd}}\right)$	$5dt$
Auxiliary Posture Maintenance		
$r_{\text{base height}}$	$\min(h_b, h_t)$	$5dt$
r_{roll}	$\lvert g_y \rvert$	$-5dt$
$r_{\text{velocity},z}$	$(v_z)^2$	$-5dt$
$r_{\text{joint limits}}$	$\sum \left[(q_{\min} - q)^+ + (q - q_{\max})^+\right]$	$-5dt$
r_{fatigue}	$\zeta \cdot \lvert \tau_d \cdot \kappa_{\text{scale}} \rvert$	$-0.05dt$
$r_{\text{joint acceleration}}$	$\ddot{q}^2$	$-1e-6dt$

IV. Growth-Based Training

Due to the highly nonlinear nature of the torque space, training a torque-based policy poses greater challenges than a position-based one, especially during early-stage exploration. To address this, we propose a biologically inspired growth mechanism that mimics animal development by progressively unlocking the robot's physical capabilities, dynamically adapting reward functions, and gradually increasing control frequency. This process facilitates more efficient policy learning while preserving stability and promoting generalization.

While related in spirit to curriculum learning [64, 65, 66, 67] and progressive learning [68, 69] approaches—which typically increase task difficulty over time—our method maintains a fixed task throughout training. Instead of staging increasingly complex goals, we focus on enhancing the agent's embodiment by gradually expanding what it is physically allowed to do. This leads to deeper exploration and reduces the risk of suboptimal shortcuts, such as exploiting a single powerful joint. Our strategy also aligns conceptually with reward scheduling techniques, where the learning signal evolves in tandem with the agent's growing capabilities [70, 71]. For instance, as the agent develops from basic contact with the ground to full stepping behavior, the reward emphasis naturally shifts to reflect the current developmental stage.

A. Implementation of the Growth Mechanism

Instead of granting the policy full access to the action space from the start of training, we propose that partially limiting the robot's abilities can promote more efficient exploration. Additionally, gradually increasing the control frequency simplifies exploration and mitigates the problem of delayed reward during the early stages of training. To unify these components, we introduce a time-dependent general scale $G(t)$, derived from the Gompertz model [72], a well-established framework for modeling growth:

$$G(t) = e^{-e^{-k \cdot (t - t_0)}}. \tag{5}$$

Here, $G(t)$ serves as the basis for dynamically adjusting training parameters. The parameters k, t, and t_0 denote the growth rate, the current training step, and the step at which the maximum growth rate occurs, respectively.

1) Torque Limits and Control Frequency Adjustment: Using $G(t)$, we dynamically update the torque limits (τ_{limit}) and control frequency (f_{policy}), which are closely linked to growth [73, 74, 75], during training. These updates enable the robot to gradually get access to its full operational capabilities:

$$\tau_{\text{limit}} = \tau_{\text{start}} + (\tau_{\text{end}} - \tau_{\text{start}}) \cdot G(t), \tag{6}$$

$$f_{policy} = f_{\text{start}} + (f_{\text{end}} - f_{\text{start}}) \cdot G(t). \tag{7}$$

Here, τ_{start} and f_{start} represent the initial torque limit and control frequency at the beginning of training, while τ_{end} and f_{end} denote their maximum values, which are reached as training progresses.

2) Dynamic Adjustment of Reward Expressions: We also leverage $G(t)$ to dynamically adjust certain reward expressions during training. This approach mirrors how animals maintain a consistent overall goal while shifting their focus across different learning stages. For instance, animals prioritize balance and upright posture during early locomotion learning, then focus on stepping, and eventually smoothen their movements. Similarly, $G(t)$ allows the robot to adapt reward priorities to align with specific training objectives. The adjusted growth-based reward r_{growth} expressions are summarized in Table II:

TABLE II
DETAILS OF OUR GROWTH-BASED REWARD TERMS

Adjusted Rewards	Calculation
$r_{\text{tracking},x}$	$\phi\left(v_x - \frac{v_{x,\max}^{\text{cmd}} + v_{x,\min}^{\text{cmd}}}{2}\right)(1 - G(t)) + \phi\left(v_x - v_x^{\text{cmd}}\right)(1 + G(t))$
$r_{\text{tracking},y}$	$\phi\left(v_y - v_y^{\text{cmd}}\right) G(t)$
$r_{\text{tracking,yaw}}$	$\phi\left(\omega_{\text{yaw}} - \omega_{\text{yaw}}^{\text{cmd}}\right) G(t)$
$r_{\text{base height}}$	$\min(h_{\text{b}}, h_{\text{t}})(1 + G(t)) - \max\left(g_x, -\min\left(0, 0.2 \cdot (1.5 - 2 \cdot G)\right)\right)$

As $G(t)$ evolves, the rewards shift focus from encouraging basic behaviors, such as forward motion, to more complex objectives like maintaining body height and tracking precise velocity commands. This progression enables the robot to transition from simple motions to refined locomotion and compliant posture control efficiently.

B. Training Details

We conduct training using Isaac Gym and the Unitree GO2 quadruped robot. This framework enables high-throughput simulation, allowing us to simulate 4096 instances of the GO2 robot in parallel on a single NVIDIA RTX 4090 GPU. We utilize Proximal Policy Optimization (PPO) to train the control policy. The hyperparameters and neural network architecture are consistent with [33], including a multilayer perceptron (MLP) with three hidden layers, whose hidden dimensions are $[512, 256, 128]$. Leveraging this framework, we achieve efficient policy learning within 20 minutes/ 3000 episodes.

The maximum episode length is set to 10 seconds. The environment resets when the robot flips over or its joint angles exceed predefined limits. The terrains include rough surfaces (with a maximum height variation of 12 cm) and slopes.

After each reset, the robot is repositioned lying flat on the ground, with varying levels of motor fatigue already applied. This random initialization enhances the generalization capability of the learned policy. Target velocity commands $[v_x^{\text{cmd}}, v_y^{\text{cmd}}, \omega_{\text{yaw}}^{\text{cmd}}]$ are sampled every 5 seconds, with ranges set to $v_x^{\text{cmd}} \in [-0.5, 1.5]$ m/s, $v_y^{\text{cmd}} \in [-0.5, 0.5]$ m/s, and $\omega_{\text{yaw}}^{\text{cmd}} \in [-1.5, 1.5]$ rad/s.

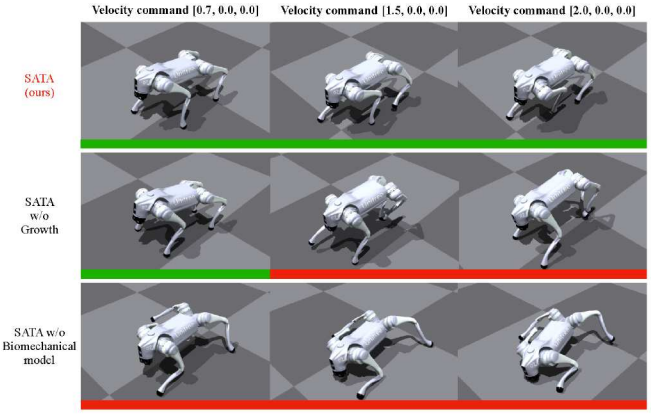

Fig. 3. Ablation study of the proposed framework, showing successful training in green and failure/premature convergence in red. SATA is compared with variants that lack the biomechanical model or the growth mechanism. Notice that without the growth model (SATA w/o Growth), the policy struggles to achieve high commanded velocities (1.5m/s), especially above the range seen during training. Without the biomechanical model (SATA w/o biomechanical model), the robot is completely unable to learn a coherent gait, instead learning to shift its feet on the floor asymmetrically.

Domain randomization is applied during training to simulate real-world variability. The specific randomization settings are as follows:

- **Added base mass:** Randomly increased by up to 5 kg.
- **Ground coefficient of friction:** Varied within [0.5, 1.25].
- **Probability to hold last actions or observations:** 10%.
- **Shifted center of mass:** Varied within [-0.2, 0.2] m along the x-axis, and [-0.1, 0.1] m along the y- and z-axes.

During training, control frequency and torque limits are progressively increased, asymptotically approaching their maximum values without fully reaching them, as defined in Eqs.6 and7. For deployment, the robot's full capacity is restored by setting these parameters to their respective maximums, f_{end} and τ_{end}. All hyperparameters related to the growth schedule and biomechanical model are summarized in Table III.

TABLE III
GROWTH SCHEDULE AND MODEL HYPERPARAMETERS

Growth Schedule					
k	0.00003	t_0	24000	τ_{start}	7.05 Nm
τ_{end}	23.5 Nm	f_{start}	100 Hz	f_{end}	200 Hz
Biomechanical Model					
κ_{scale}	5.0	γ	0.6	β	0.9

V. EXPERIMENTS

A. Simulation Experiments

1) Ablation Study: To evaluate the contribution of each component of our approach, we compare the performance of the complete framework (SATA) with variants that remove the biomechanical model (SATA w/o biomechanical model) or the growth mechanism (SATA w/o Growth).

As shown in Fig. 3, the biomechanical model plays a critical role in enabling natural and stable locomotion. When

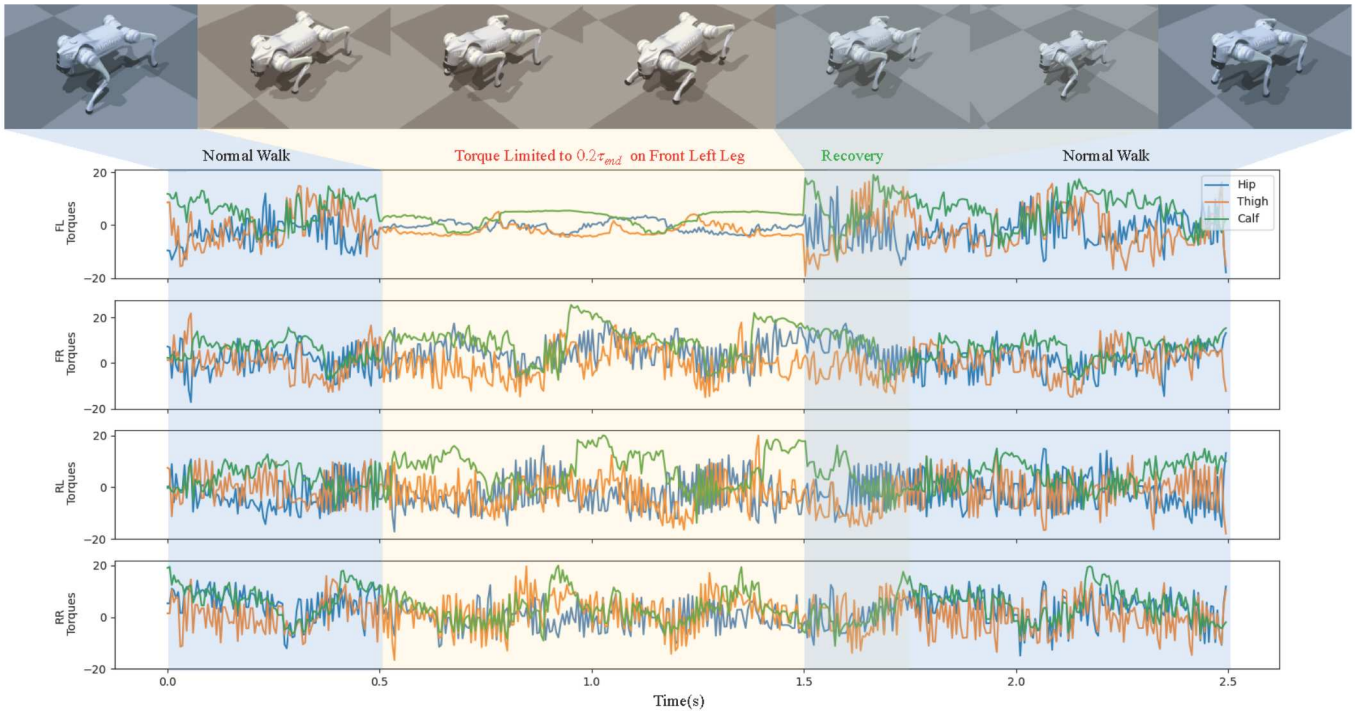

Fig. 4. Response to a sudden torque limitation on the front left leg (at $t = 0.5\,\mathrm{s}$). During this disturbance ($0.5\,\mathrm{s} < t < 1.5\,\mathrm{s}$), the robot dynamically compensates using other legs, and rapidly recovers once the torque is restored ($1.5\,\mathrm{s} < t < 1.75\,\mathrm{s}$).

this biomechanical model is removed, the robot converges to unnatural gaits, such as three-legged support patterns, which reduce stability and limit energy efficiency. This highlights the importance of the biomechanical model and feedback mechanisms in smoothing motion commands and preventing suboptimal convergence.

On the other hand, the inclusion of the growth mechanism leads to higher early-stage training efficiency and shows better generalization when tracking out-of-distribution (OOD) velocity commands. As demonstrated in Fig. 5a, SATA significantly outperforms SATA w/o growth in early stages of training, demonstrating the impact of this mechanism in early stage exploration. Moreover, when comparing the cumulative reward of both scenarios under OOD velocity commands ($v_x = 1.8m/s$) as in Fig. 5b, we can see that our method outperforms SATA w/o growth, demonstrating the impact of the growth mechanism on policy generalization. Upon closer inspection, as in Fig. 3, a risky gait emerges as the policy is commanded to the highest command seen in training and in OOD scenarios, leading to unstable tracking of velocity commands.

In particular, these results suggest that the complementary roles of the biomechanical model help ensure stable and natural motion, and the growth mechanism enhance the policy's adaptability and robustness under diverse conditions.

2) Robustness to Single-Leg Failure: In this experiment, we simulate the failure of a single leg by abruptly reducing the maximum torque of its motor to 20% of its original capacity ($0.2\tau_{\text{end}}$). By doing so, we validate the robustness of our policy during asymmetric conditions.

As shown in Fig. 4, when the front left leg's torque output

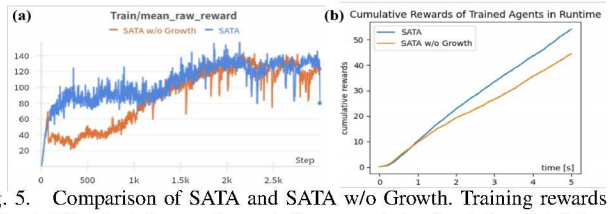

Fig. 5. Comparison of SATA and SATA w/o Growth. Training rewards (a), without G(t) adaptation, and cumulative rewards in simulation test (b), when commanded to run at 1.8 m/s (slightly OOD).

is limited, the other legs adaptively increase their force output to stabilize the robot's posture and prevent a collapse. This dynamic redistribution of effort ensures continuous and stable locomotion even under single leg failures. Once the torque capacity of the front left leg is restored, the robot reactively transitions back to its normal walking gait, demonstrating the efficiency of the adaptive feedback mechanism in handling and recovering from localized perturbations.

TABLE IV
PERFORMANCE COMPARISON BETWEEN OUR METHOD AND BASELINES ACROSS DIFFERENT ROBUSTNESS TESTS (5 TRIALS PER TEST)

	Sideway Pushing	Soft Terrain	Tunnel	Vertical Stomp
SATA (Ours)	**100%**	**100%**	**100%**	**100%**
WalkTheseWay	20%	20%	20%	20%
Vanilla Position-based Policy	80%	40%	0%	20%
DeCAP (Pure Torque)	0%	0%	0%	60%
DeCAP (Position-assisted)	80%	40%	0%	80%
Unitree Built-in MPC	80%	100%	0%	80%

B. Lab Level Experiments

To validate the effectiveness of our approach, we deployed it on a Unitree Go2 quadruped robot in real-world scenarios. We also compared its performance against several baseline methods, including Unitree's built-in, MPC-based controller,

Fig. 6. Pushing the robot to its left (a) and right (b), and manually lifting its legs (c). In all those cases, the controller did not start trotting nor generate overreacting/hazardous motions.

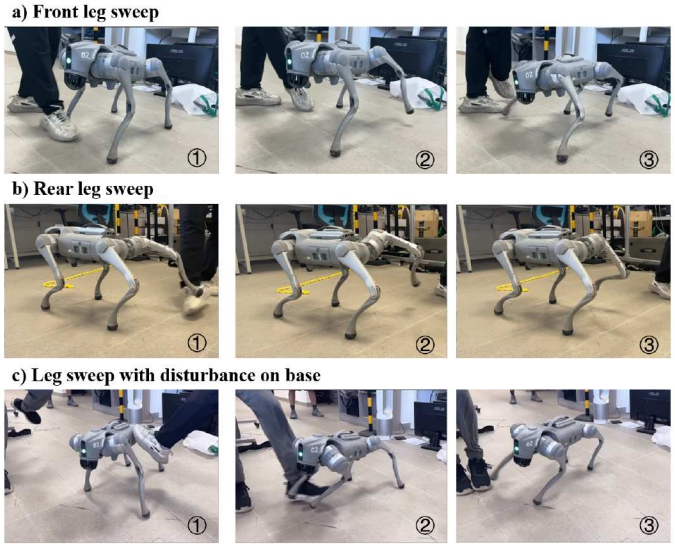

Fig. 7. Leg disturbance test with a) Backward sweep of the front legs, b) Backward sweep of the back legs, and c) Forward sweep of the front legs right after kicking its body. In a), the rear left leg also lifts up to help balance the whole body, while in b) the robot simply shifts its weight to the other three feet in response to the sweep.

a vanilla learning position-based policy, a well-known learning position-based policy, WalkTheseWay [9], as well as a torque-based policy, DeCAP [11]. Experimental results demonstrated that our approach exhibited robust performance across various scenarios, including handling unexpected disturbances and navigating unseen environments not encountered during training. In multiple evaluations, as summarized in Table IV, our method consistently outperformed the baseline approaches across all robustness tests (please refer to our associated video for details).

Contrary to the common perception that torque-based methods suffer from a significant sim-to-real gap, our approach achieved zero-shot transfer without any fine-tuning and demonstrated highly stable operation over extended deployment periods. To assess its compliance and adaptability, we conducted a series of external disturbance tests in a controlled lab environment to observe and compare the controller's responses against baseline methods. In the first subsection, we illustrate the compliance of our method during **human-robot interactions**, while Sections V-B2 and V-B3 highlight its robustness against **external disturbances**.

1) Safety for Human-Robot Interaction: For robotic manipulators, compliance is critical for ensuring human safety during collaborative tasks. A compliant robotic arm can be easily pushed by a person and will stop applying excessive force upon contact, preventing injuries instead of forcefully moving to a predetermined position. Similarly, compliance

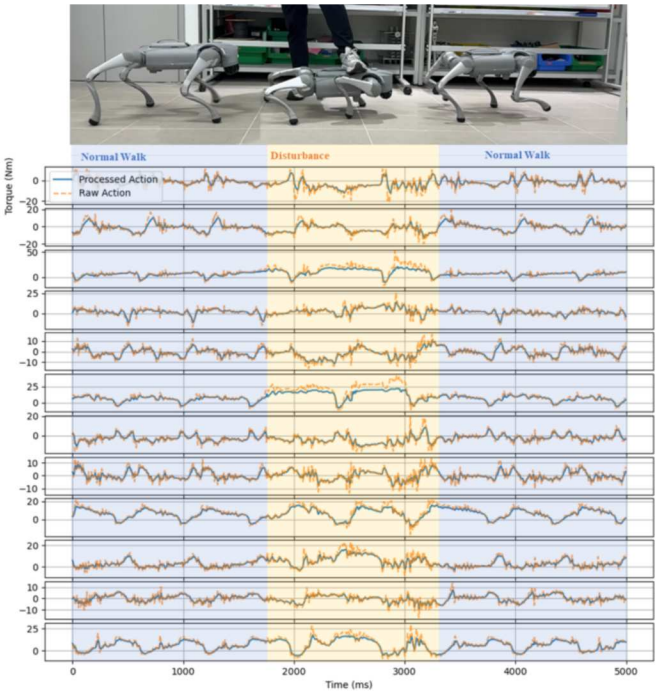

Fig. 8. Walking under external downward forces/presses. The blue line is the actual torque command given to the motors after the processing done by our biomechanical model, while the orange dotted line is the raw action output of the policy. The robot can continue progressing forward even when an external force press it to the ground. Note that the biomechanical model is crucial in ensuring the safety of the robot especially during these disturbances, limiting excessive torque output and mitigating oscillations.

in quadrupedal robots is essential for safe interaction with humans. This section focuses on evaluating the passive compliance of our controller and its interaction capabilities with humans.

In standing posture, our torque-based controller allows the robot to respond naturally to human-applied forces, adjusting its body posture without unnecessary stepping motions. The robot only takes corrective steps when balance is at risk. Fig. 6 illustrates the level of compliance the robot can achieve during human robot interaction. These results significantly surpass the performance of position-based controllers, which exhibit a very stiff behavior and are nearly unable to be displaced without stepping.

2) Impact on Legs: Compliance also plays a crucial role in responding to localized external disturbances. To test this, we placed our robot in a standing posture ($v^{cmd} = 0$) while its legs were subjected to forward and lateral sweeps with sufficient force to lift up its foot. As demonstrated in Fig. 7, the robot's controller exhibited robust performance, successfully resisting these disturbances across all four legs without overreacting. Furthermore, even when additional external impacts were applied to other parts of the body, the controller consistently enabled the robot to regain balance through a series of smooth, adaptive adjustments, quickly regaining balance with a side step.

3) Kicking and Stomping: Most quadrupedal controllers demonstrate robust responses to kicks. However, the impact limit for these controllers varies. Some struggles with recovery

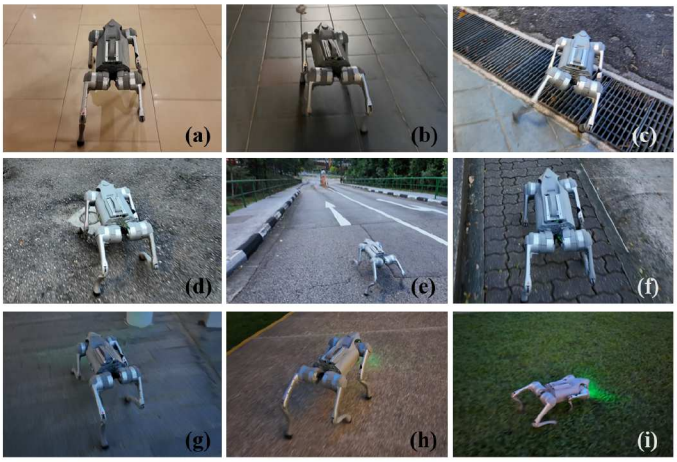

Fig. 9. Stability test of our proposed controller over a long (1.2km) route, which covers tiled floor indoors (a,b), rough hard unstructured road (c,d), high slopes (e), pedestrian paths (f,g,h), and soft lawn (i). The whole route was completed without human correction, beyond manual adjustment of the robot's heading to chart its course.

from downward forces applied directly from above, while others perform poorly from horizontal impacts that can flip the robot over. In contrast, our controller exhibits remarkable recovery capabilities to these disturbances. Fig. 8 shows how our approach handles a downward stomp during locomotion, with its biomechanical model helping to ensure safe motor commands. Furthermore, we note that our method allows the robot to successfully start operating from arbitrary configurations, such as lying flat or standing upright, similar to animals. This capability is rarely observed in position-based methods or traditional control approaches, which typically require the robot to stand up first through predefined instructions before operation.

C. Out-of-distribution Environments

Fig. 9 shows a very long (1.2km) walk using our controller, passing through different terrains without any human correction, except for manually controlling the robot's heading to chart its course. Generally, position-based quadrupedal controllers perform well on various hard, unstructured surfaces. However, significant challenges remain when navigating soft or slippery terrain. As noted in [3], additional modifications to the controller or extensive domain randomization are often required to ensure robust real-world deployment on such surfaces. On the other hand, torque-based controllers leverage a more direct understanding of the robot's dynamics and terrain interactions, enabling a more robust response to those environments. The following section covers several challenging and out-of-distribution terrains, including height constraint, or soft and slippery grounds, showcasing the generalizability and adaptability of our approach.

1) Squeezing into a Tunnel: To evaluate the performance of our controller in height-constrained scenarios, we designed a tunnel traversal experiment in which the minimum tunnel height was approximately 30 cm. Unlike baseline methods which can hardly be compressed by vertical force, our approach is able to passively adapt to this situation. As shown in Fig. 10, the robot was compressed to approximately half of its normal height after bumping into the tunnel ceiling, and was able to move forward while staying in this compressed state. Despite the fact that this scenario was never seen during training, and that no external height command was given, the controller autonomously adopted a crawling gait, allowing the robot to continue moving forward without getting stuck. Upon exiting the tunnel, the robot naturally returned to its upright posture and resumed its normal walking gait without requiring any external intervention.

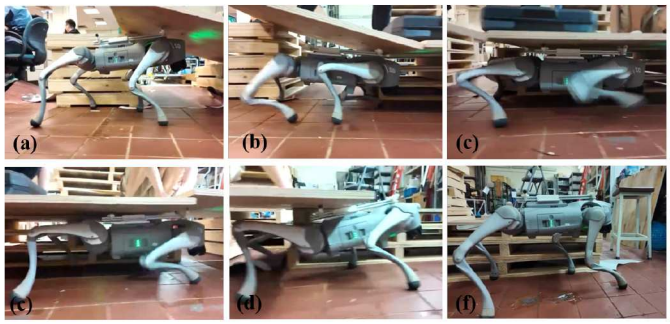

Fig. 10. Locomotion through a height-constrained space. Notably, no height command or additional modifications were made to the robot or the policy, except for the addition of passive wheels on top of the robot's body to reduce friction from the tunnel ceiling.

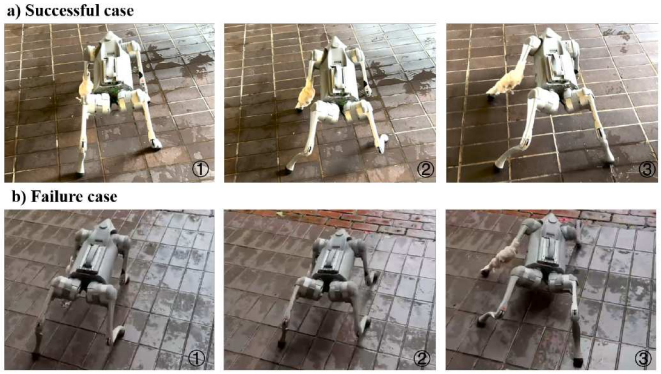

Fig. 11. Locomotion on wet slippery surfaces, showing both success (a) and failure (b). Even when the foot of the robot slips and fall down in failure cases, the controller still does not exhibit wild, unsafe motions.

2) Slippery Surfaces: Figure 11 demonstrates the robot's performance on a wet and slippery surface. In Figure 11a, after experiencing a slip during locomotion, the robot quickly recovers its standing posture. Our controller dynamically adjusts the motor output to stabilize the robot without predefined recovery motions. In contrast, Figure 11b shows a failure case, where the robot is given an abrupt command on the slippery surface. Note that the controller is still active with the robot applying torque through the thigh joints in an attempt to recover balance, yet no abrupt or flailing motion is produced. This behavior contrasts sharply with position-based controllers, where the thigh and calf joint would typically be retracted abruptly, often leading to tipping or other destabilizing outcomes. The torque-based controller's ability to exert gradual and compliant force enhances minimizes the aggressive reactions that could compromise stability.

3) Soft Terrain: In addition to slippery surfaces, deformable terrain presents significant challenges for position-based controllers. To evaluate performance, we conducted an experiment

Fig. 12. Locomotion on 10cm thick soft mattress with a velocity command of 0.8m/s. Our robot stops with the right most posture when the velocity command is finally set to 0.

where the robot traversed a soft mattress approximately 10 cm thick. As shown in Figure 12, a position-based reinforcement learning controller struggled to cross the terrain effectively, often failing to maintain forward momentum or balance. In contrast, our torque-based controller demonstrated robust performance, successfully navigating the deformable surface without any additional modification. This highlights the advantage of torque control in dynamically adapting to terrain variability, leveraging compliant interaction with the environment for enhanced stability and maneuverability.

By directly regulating motor torques, our policy allows the robot to exhibit greater compliance and stability, reducing the dependency on domain randomization or manual tuning commonly needed for position-based approaches. We believe these advantages highlight the potential of torque-based control for robust real-world deployments in diverse environments.

VI. Limitations

A. Challenges in Posture Maintenance

Our current control strategy is torque-based, which offers enhanced compliance, enabling the robot to better adapt to impacts from various directions. However, this compliance also comes at the cost of posture maintenance. For instance, when carrying a medium-weight payload (5 kg), a scenario that position-based control systems can easily handle, our robot is not able to maintain its body height during walking. This often results in the robot's calfs coming into contact with the ground, which risks damaging the robot. To address this limitation, we believe that further optimization of the biomechanical model may be necessary, to allow the robot to better understand and adapt (comply with, or safely resists to) different types of external forces. For example, introducing tunable parameters or introducing antagonistic muscle pairs could enable the robot to dynamically adjust the stiffness of its virtual muscles, thereby enhancing its posture maintenance capabilities when required.

B. Gait Learning Limitations

Compared to position-based control systems, which can learn and execute a diverse range of gaits and generate relatively more dynamic gaits, such as trot, amble, or gallop, our torque-based policy has so far only successfully learned a basic walking gait. Even at high speeds, such as 2.0 m/s, the robot continues to utilize a walking gait pattern, merely increasing its stride length. We believe that this limitation may hinder the robot's performance in highly dynamic scenarios and could lead to increased energy consumption. In future work, we will seek ways to first manually guide the learning of different gaits, and then look for mechanisms that may allow them to emerge from the training (alongside methods to stably switch between them), once again by drawing inspiration from biological principles and structures.

VII. Conclusion

In this work, we presented SATA, a bio-inspired torque-based learning framework for quadrupedal locomotion, aimed at achieving safer and more compliant robot behaviors. By incorporating biomechanical principles and adaptive learning mechanisms inspired by animal locomotion, SATA enhances compliance, adaptability, and generalization, enabling robots to interact more effectively with diverse and challenging environments. Our framework leverages a muscle-like model to smoothen motion commands and mitigate suboptimal behaviors, alongside a growth-inspired training mechanism that progressively unlocks robot capabilities and improves early-stage exploration efficiency. Experimental results demonstrate SATA's robustness in handling a variety of scenarios, including unseen velocity commands, soft or slippery terrains, and real-world disturbances such as sudden torque limitations and external impacts. Moreover, SATA achieves zero-shot sim-to-real transfer, removing the need for additional hardware fine-tuning while maintaining safe and robust locomotion.

Our future work will extend SATA's capabilities to more complex terrains and agile behaviors, further improving its adaptability in unstructured/dynamic environments. We further plan to explore hybrid learning strategies that integrate adaptive muscle-like actuation, to further boost compliance and robustness. We envision that these advancements will bring us ever closer to deploying safe and versatile torque-based controllers in real-world scenarios.

VIII. Acknowledgments

This work was supported by the Singapore Ministry of Education Academic Research Fund Tier 1, as well as the National Research Foundation, Singapore (NRF), the Maritime and Port Authority of Singapore (MPA), the Singapore Maritime Institute (SMI) under its Maritime Transformation Programme (Project No. SMI-2022-MTP-01), and the Swiss National Science Foundation, under Grant 200021_197237.

References

[1] Ying-Sheng Luo, Jonathan Hans Soeseno, Trista Pei-Chun Chen, and Wei-Chao Chen. Carl: Controllable agent with reinforcement learning for quadruped locomotion. *ACM Transactions on Graphics (TOG)*, 39(4): 38–1, 2020.

[2] Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. *arXiv preprint arXiv:1804.10332*, 2018.

[3] Suyoung Choi, Gwanghyeon Ji, Jeongsoo Park, Hyeongjun Kim, Juhyeok Mun, Jeong Hyun Lee, and Jemin Hwangbo. Learning quadrupedal locomotion on deformable terrain. *Science Robotics*, 8(74):eade2256, 2023.

[4] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5 (47):eabc5986, 2020.

[5] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science robotics*, 7(62):eabk2822, 2022.

[6] Biao Hu, Shibo Shao, Zhengcai Cao, Qing Xiao, Qunzhi Li, and Chao Ma. Learning a faster locomotion gait for a quadruped robot with model-free deep reinforcement learning. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1097–1102. IEEE, 2019.

[7] Vassilios Tsounis, Mitja Alge, Joonho Lee, Farbod Farshidian, and Marco Hutter. Deepgait: Planning and control of quadrupedal gaits using deep reinforcement learning. *IEEE Robotics and Automation Letters*, 5(2): 3699–3706, 2020.

[8] Taehei Kim and Sung-Hee Lee. Quadruped locomotion on non-rigid terrain using reinforcement learning. *arXiv preprint arXiv:2107.02955*, 2021.

[9] Gabriel B Margolis and Pulkit Agrawal. Walk these ways: Tuning robot control for generalization with multiplicity of behavior. In *Conference on Robot Learning*, pages 22–31. PMLR, 2023.

[10] Mohsen Sombolestan and Quan Nguyen. Adaptive force-based control of dynamic legged locomotion over uneven terrain. *IEEE Transactions on Robotics*, 2024.

[11] Shivam Sood, Ge Sun, Peizhuo Li, and Guillaume Sartoretti. Decap: Decaying action priors for accelerated imitation learning of torque-based legged locomotion policies. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2809–2815. IEEE, 2024.

[12] Shuxiao Chen, Bike Zhang, Mark W Mueller, Akshara Rai, and Koushil Sreenath. Learning torque control for quadrupedal locomotion. In *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, pages 1–8. IEEE, 2023.

[13] Chun Y Seow. Hill's equation of muscle performance and its hidden insight on molecular mechanisms. *Journal of General Physiology*, 142(6):561–573, 2013.

[14] Roger M Enoka and Jacques Duchateau. Muscle fatigue: what, why and how it influences muscle function. *The Journal of physiology*, 586(1):11–23, 2008.

[15] Guillaume Bellegarda, Yiyu Chen, Zhuochen Liu, and Quan Nguyen. Robust high-speed running for quadruped robots via deep reinforcement learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10364–10370. IEEE, 2022.

[16] Guillaume Bellegarda, Chuong Nguyen, and Quan Nguyen. Robust quadruped jumping via deep reinforcement learning. *Robotics and Autonomous Systems*, 182: 104799, 2024.

[17] Liang Ren, Chunlei Wang, Ya Yang, and Zhiqiang Cao. A learning-based control approach for blind quadrupedal locomotion with guided-drl and hierarchical-drl. In *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 881–886. IEEE, 2021.

[18] Yanran Ding, Abhishek Pandala, and Hae-Won Park. Real-time model predictive control for versatile dynamic motions in quadrupedal robots. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8484–8490. IEEE, 2019.

[19] Donghyun Kim, Jared Di Carlo, Benjamin Katz, Gerardo Bledt, and Sangbae Kim. Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control. *arXiv preprint arXiv:1909.06586*, 2019.

[20] Michael Neunert, Markus Stäuble, Markus Giftthaler, Carmine D Bellicoso, Jan Carius, Christian Gehring, Marco Hutter, and Jonas Buchli. Whole-body nonlinear model predictive control through contacts for quadrupeds. *IEEE Robotics and Automation Letters*, 3 (3):1458–1465, 2018.

[21] Ruben Grandia, Fabian Jenelten, Shaohui Yang, Farbod Farshidian, and Marco Hutter. Perceptive locomotion through nonlinear model-predictive control. *IEEE Transactions on Robotics*, 39(5):3402–3421, 2023.

[22] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018.

[23] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.

[24] Michel Aractingi, Pierre-Alexandre Léziart, Thomas Flayols, Julien Perez, Tomi Silander, and Philippe Souères. Controlling the solo12 quadruped robot with deep reinforcement learning. *scientific Reports*, 13(1): 11945, 2023.

[25] Yuxiang Yang, Ken Caluwaerts, Atil Iscen, Tingnan Zhang, Jie Tan, and Vikas Sindhwani. Data efficient reinforcement learning for legged robots. In *Conference on Robot Learning*, pages 1–10. PMLR, 2020.

[26] Arthur Allshire, Roberto Martín-Martín, Charles Lin, Shawn Manuel, Silvio Savarese, and Animesh Garg. Laser: Learning a latent action space for efficient reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6650–6656. IEEE, 2021.

[27] MyeongSeop Kim, Jung-Su Kim, and Jae-Han Park. Automated hyperparameter tuning in reinforcement learning for quadrupedal robot locomotion. *Electronics*, 13(1):

116, 2023.

[28] Shangke Lyu, Xin Lang, Han Zhao, Hongyin Zhang, Pengxiang Ding, and Donglin Wang. Rl2ac: Reinforcement learning-based rapid online adaptive control for legged robot robust locomotion. In *Proceedings of the Robotics: Science and Systems*, 2024.

[29] Jonas Buchli, Mrinal Kalakrishnan, Michael Mistry, Peter Pastor, and Stefan Schaal. Compliant quadruped locomotion over rough terrain. In *2009 IEEE/RSJ international conference on Intelligent robots and systems*, pages 814–820. IEEE, 2009.

[30] Andrea Calanca, Riccardo Muradore, and Paolo Fiorini. A review of algorithms for compliant control of stiff and fixed-compliance robots. *IEEE/ASME transactions on mechatronics*, 21(2):613–624, 2015.

[31] Donghyeon Kim, Glen Berseth, Mathew Schwartz, and Jaeheung Park. Torque-based deep reinforcement learning for task-and-robot agnostic learning on bipedal robots using sim-to-real transfer. *IEEE Robotics and Automation Letters*, 2023.

[32] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[33] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100. PMLR, 2022.

[34] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning, 2022. URL https://arxiv.org/abs/2109.11978.

[35] Guoyu Zuo, Yong Wang, Daoxiong Gong, and Shuangyue Yu. Learning quadrupedal locomotion on tough terrain using an asymmetric terrain feature mining network. *Applied Intelligence*, 54(22):11547–11563, 2024.

[36] Yiyu Chen and Quan Nguyen. Learning agile locomotion and adaptive behaviors via rl-augmented mpc. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11436–11442. IEEE, 2024.

[37] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

[38] Siddhant Gangapurwala, Mathieu Geisert, Romeo Orsolino, Maurice Fallon, and Ioannis Havoutis. Rloc: Terrain-aware legged locomotion using reinforcement learning and optimal control. *IEEE Transactions on Robotics*, 38(5):2908–2927, 2022.

[39] Qingfeng Yao, Jilong Wang, Donglin Wang, Shuyu Yang, Hongyin Zhang, Yinuo Wang, and Zhengqing Wu. Hierarchical terrain-aware control for quadrupedal locomotion by combining deep reinforcement learning and optimal control. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4546–4551. IEEE, 2021.

[40] Zhitong Zhang, Honglei An, Qing Wei, and Hongxu Ma. Learning-based model predictive control for quadruped locomotion on slippery ground. In *2022 4th International Conference on Control and Robotics (ICCR)*, pages 47–52. IEEE, 2022.

[41] Zhitong Zhang, Xu Chang, Hongxu Ma, Honglei An, and Lin Lang. Model predictive control of quadruped robot based on reinforcement learning. *Applied Sciences*, 13 (1):154, 2022.

[42] Amir H Abdi, Masoud Malakoutian, Thomas Oxland, and Sidney Fels. Reinforcement learning for high-dimensional continuous control in biomechanics: an intro to artisynth-rl. *arXiv preprint arXiv:1910.13859*, 2019.

[43] Wenjuan Ouyang, Haozhen Chi, Jiangnan Pang, Wenyu Liang, and Qinyuan Ren. Adaptive locomotion control of a hexapod robot via bio-inspired learning. *Frontiers in Neurorobotics*, 15:627157, 2021.

[44] Joseph Humphreys and Chengxu Zhou. Learning to adapt: Bio-inspired gait strategies for versatile quadruped locomotion. *arXiv preprint arXiv:2412.09440*, 2024.

[45] Jiayu Wang, Chuxiong Hu, and Yu Zhu. Cpg-based hierarchical locomotion control for modular quadrupedal robots using deep reinforcement learning. *IEEE Robotics and Automation Letters*, 6(4):7193–7200, 2021.

[46] Tianqi Wei and Barbara Webb. A bio-inspired reinforcement learning rule to optimise dynamical neural networks for robot control. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 556–561. IEEE, 2018.

[47] Xinyu Zhang, Zhiyuan Xiao, Qingrui Zhang, and Wei Pan. Synloco: Synthesizing central pattern generator and reinforcement learning for quadruped locomotion. *arXiv preprint arXiv:2310.06606*, 2023.

[48] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. *arXiv preprint arXiv:2004.00784*, 2020.

[49] Zipeng Fu, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Minimizing energy consumption leads to the emergence of gaits in legged robots. *arXiv preprint arXiv:2111.01674*, 2021.

[50] Guillaume Bellegarda and Auke Ijspeert. CPG-RL: Learning central pattern generators for quadruped locomotion. *IEEE Robotics and Automation Letters*, 7(4): 12547–12554, 2022.

[51] Ge Sun, Milad Shafiee, Peizhuo Li, Guillaume Bellegarda, Auke Ijspeert, and Guillaume Sartoretti. Learning-based hierarchical control: Emulating the central nervous system for bio-inspired legged robot locomotion. *arXiv preprint arXiv:2404.17815*, 2024.

[52] Zhengmao He, Kun Lei, Yanjie Ze, Koushil Sreenath, Zhongyu Li, and Huazhe Xu. Learning visual quadrupedal loco-manipulation from demonstrations. *arXiv preprint arXiv:2403.20328*, 2024.

[53] Deepali Jain, Atil Iscen, and Ken Caluwaerts. Hierarchical reinforcement learning for quadruped locomotion. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7551–7557. IEEE, 2019.

[54] Deepali Jain, Atil Iscen, and Ken Caluwaerts. From pixels to legs: Hierarchical learning of quadruped locomotion. *arXiv preprint arXiv:2011.11722*, 2020.

[55] abc Botterman and abc Gonyea. Gradation of isometric tension by different activation rates in motor units of cat flexor carpi radialis muscle. *Journal of neurophysiology*, 56(2):494–506, 1986.

[56] Arnault H Caillet, Andrew TM Phillips, Dario Farina, and Luca Modenese. Motoneuron-driven computational muscle modelling with motor unit resolution and subject-specific musculoskeletal anatomy. *PLOS Computational Biology*, 19(12):e1011606, 2023.

[57] Damien M Callahan, Brian R Umberger, and Jane A Kent-Braun. A computational model of torque generation: neural, contractile, metabolic and musculoskeletal components. *PloS one*, 8(2):e56013, 2013.

[58] Vahid Hassani, Tegoeh Tjahjowidodo, and Thanh Nho Do. A survey on hysteresis modeling, identification and control. *Mechanical systems and signal processing*, 49 (1-2):209–233, 2014.

[59] Syn Schmitt, Michael Günther, and Daniel FB Häufle. The dynamics of the skeletal muscle: A systems biophysics perspective on muscle modeling with the focus on hill-type muscle models. *GAMM-Mitteilungen*, 42(3): e201900013, 2019.

[60] F Romero and FJ Alonso. A comparison among different hill-type contraction dynamics formulations for muscle force estimation. *Mechanical Sciences*, 7(1):19–29, 2016.

[61] Olaf Till, Tobias Siebert, Christian Rode, and Reinhard Blickhan. Characterization of isovelocity extension of activated muscle: a hill-type model for eccentric contractions and a method for parameter determination. *Journal of theoretical biology*, 255(2):176–187, 2008.

[62] TW Boonstra and PJ Beek. Fatigue-related changes in motor-unit synchronization of quadriceps muscles within and across legs. *Journal of Electromyography and Kinesiology*, 18(5):717–731, 2008.

[63] Jeffrey C Cowley and Deanna H Gates. Inter-joint coordination changes during and after muscle fatigue. *Human Movement Science*, 56:109–118, 2017.

[64] Bangyu Qin, Yue Gao, and Yi Bai. Sim-to-real: Six-legged robot control with deep reinforcement learning and curriculum learning. In *2019 4th International Conference on Robotics and Automation Engineering (ICRAE)*, pages 1–5. IEEE, 2019.

[65] Wenhao Yu, Greg Turk, and C Karen Liu. Learning symmetric and low-energy locomotion. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018.

[66] Sicen Li, Gang Wang, Yiming Pang, Panju Bai, Shihao Hu, Zhaojin Liu, Liquan Wang, and Jiawei Li. Learning agility and adaptive legged locomotion via curricular hindsight reinforcement learning. *Scientific Reports*, 14 (1):28089, 2024.

[67] Taisuke Kobayashi and Toshiki Sugino. Reinforcement learning for quadrupedal locomotion with design of continual–hierarchical curriculum. *Engineering Applications of Artificial Intelligence*, 95:103869, 2020.

[68] Glen Berseth, Cheng Xie, Paul Cernek, and Michiel Van de Panne. Progressive reinforcement learning with distillation for multi-skilled motion control. *arXiv preprint arXiv:1802.04765*, 2018.

[69] Yike Li, Yunzhe Tian, Endong Tong, Wenjia Niu, and Jiqiang Liu. Robust reinforcement learning via progressive task sequence. In *IJCAI*, pages 455–463, 2023.

[70] Martin Riedmiller and Jost Springenberg. Learning by playing solving sparse reward tasks from scratch. In *International conference on machine learning*, pages 4344–4353. PMLR, 2018.

[71] Aleksandra Faust and Dar Mehta. Evolving rewards to automate reinforcement learning. *arXiv preprint arXiv:1905.07628*, 2019.

[72] A Ardelean and E Suteu. The estimation of the growth curve at dog. *Bull Univ Agric Sci Vet Med Cluj-Napoca-Vet Med*, 63:175–181, 2005.

[73] Lars Larsson and Jan Karlsson. Muscle strength and speed of movement in relation to age and muscle morphology. *Journal of Applied Physiology*, 46(3):451–456, 1979.

[74] abc FH Gage. Structural plasticity of the adult brain. *Dialogues in clinical neuroscience*, 6(2):135–141, 2004.

[75] abc Stampanoni Bassi and abc Buttari. Synaptic plasticity shapes brain connectivity: implications for network topology. *International journal of molecular sciences*, 20(24):6193, 2019.