# A Lyapunov-Guided Post-Action Shield for Stability-Aware Deep Reinforcement Learning

# A Lyapunov-Guided Post-Action Shield for Stability-Aware Deep Reinforcement Learning

Hussain Kahil[1], Petri Välisuo[1], Mohammed Elmusrati[1]

***Abstract*— This paper proposes a Lyapunov-guided post-action shielding mechanism for deep reinforcement learning (DRL) controllers under bounded actuation disturbances. In addition, an energy-safety requirement is formulated as a one-step energy threshold constraint that keeps the predicted next-state energy proxy within a prescribed limit, using a bounded-disturbance worst-case check. Simulation results show that the proposed mechanism substantially reduces constraint violations under actuation noise while preserving the nominal policy behavior whenever possible.**

***Index Terms*— safe reinforcement learning, Lyapunov stability, post-action shielding, energy threshold constraints, learned controller robustness.**

## I. Introduction

Deep reinforcement learning (DRL) has recently attracted significant attention as a control system approach [1]. It can be interpreted as an adaptive, optimal, learning-based control framework whose theoretical foundations originate from the computer science and machine learning communities [2]. By interacting directly with the environment (the plant in control theory), DRL algorithms are capable of learning control policies without requiring explicit system models. This model-free learning capability allows DRL methods to overcome several fundamental limitations associated with traditional model-based control approaches. As a result, DRL-based control has emerged as an active research topic across a wide range of application domains such as smart grids optimization [3], robotics control and navigation [4] and energy systems management [5]. Despite the promising performance gains demonstrated in recent studies, the adoption of DRL in safety-critical control systems remains challenging. Unlike conventional control methods, DRL relies on policy representations parameterized by deep neural networks, which introduces nonlinearity, approximation errors, and stochasticity into the closed-loop system. These characteristics raise significant concerns regarding stability and safety during both training and deployment of DRL algorithms. In control theory, stability and safety certification have long been central research themes, traditionally addressed through mathematically rigorous frameworks such as Lyapunov stability theory which provides sufficient conditions for stability by analyzing the evolution of an energy-like function along system trajectories [6], and Control Barrier Function (CBF) which enforce safety constraints by restricting the system state to remain within predefined safe sets [7]. Beyond the widely adopted Lyapunov- and barrier-based methods, alternative frameworks have been developed to address uncertainty, safety, and robustness in nonlinear and safety-critical systems. These include robust control techniques, such as $\mathcal{H}_\infty$ and $\mu$-synthesis [8], which ensure performance and stability under bounded uncertainties; passivity-based and energy-based control methods [9], which exploit physical energy dissipation properties to guarantee stability; invariant set and viability theory approaches [10], which characterize safe operating regions through forward-invariant sets; and Reachable Predictive Control (RPC) [11] and Model Predictive Control (MPC) [12] frameworks, where safety is ensured by solving constrained optimization or Hamilton–Jacobi reachability problems online. These methods have played a central role in the design and analysis of nonlinear, adaptive, and safety-critical control systems. However, these methods are generally difficult to integrate with learning-based control frameworks such as DRL, due to their reliance on explicit system models, online optimization, or conservative assumptions.

Motivated by their strong theoretical guarantees and easy to integrate with many control algorithms, recent research efforts have focused on incorporating Lyapunov-based and constraint-based concepts into DRL frameworks to enforce stability and safety during both learning and deployment. Several Lyapunov-inspired DRL approaches have been proposed, including Lyapunov-constrained policy optimization [13], safe action projection methods [14], Lyapunov candidate value functions [15], stability-aware critic formulations [16], reward shaping via Lyapunov-based penalties [17]. Moreover, hybrid formulations that combine Lyapunov and barrier functions [18] have further enabled the simultaneous treatment of stability and safety objectives. Most Lyapunov-based DRL methods employ a fixed Lyapunov function $V(x)$ (or a learned Lyapunov critic $V_\phi(x)$) and incorporate it directly into the training loop. In these approaches, any stability/safety claim is tied to the chosen Lyapunov representation and to assumptions about how uncertainty enters the dynamics. In addition, many implementations do not explicitly distinguish between (i) the *disturbance model* used to perturb the system during training/evaluation and (ii) the *robustness model* assumed when enforcing Lyapunov decrease. As a result, it can be difficult to attribute performance gains to the controller/shield itself rather than to differences in disturbance generation or bounding.

In this work, we propose a *Lyapunov post-shielding* mechanism for DRL-based controllers operating under bounded actuator disturbances. A vanilla DRL Proximal Policy Op-

[1]School of Technology and Innovations, University of Vaasa, Vaasa, Finland
{hussain.kahil, petri.valisuo, mohammed.elmusrati}@uwasa.fi
Corresponding author: Hussain Kahil.

timization (PPO) is trained *without* noise injection and *without* Lyapunov constraints. During deployment, we introduce *stochastic physical actuation noise* inside the environment, while the policy continues to output an unconstrained nominal action $u_{\text{nom}}$. Stability is enforced only at execution time via a Lyapunov post-filter that minimally adjusts $u_{\text{nom}}$ into the applied command $u_{\text{cmd}}$ to satisfy a one-step Lyapunov decrease condition under a clearly specified disturbance bound. In addition, we impose an explicit *energy-safety* requirement as a discrete-time one-step energy bound by keeping the predicted next-state energy proxy within a prescribed limit, and we robustify the enforcement of this constraint under the same bounded disturbance model. This separation enables a controlled robustness assessment: the disturbance model and sampling procedure are explicitly defined, and the safety/stability mechanisms are applied at runtime without modifying the learned policy or its training process. The remainder of this paper is organized as follows: Section II introduces the fundamental concepts and notation used throughout this study. Section III presents the architecture and theoretical formulation of the proposed plug-in Lyapunov post-shielding mechanism. Section IV provides a proof-of-concept demonstration illustrating the effectiveness of the proposed approach using the Gymnasium InvertedPendulum-v5 environment. Finally, Section V summarizes the main findings of this work and outlines directions for future research.

## II. Preliminaries

### A. System Dynamics and Equilibrium Concepts

Consider a nonlinear continuous-time dynamical system of the form:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) \tag{1}$$

where $x(t) \in \mathbb{R}^n$ denotes the system state and $u(t) \in \mathbb{R}^m$ is the control input. The functions $f$ and $g$ are assumed to be locally Lipschitz, ensuring the standard conditions for existence and uniqueness of system trajectories.

An equilibrium point $x^* \in \mathbb{R}^n$ of system (1) corresponding to a constant control input $u^* \in \mathbb{R}^m$ is defined as a state satisfying:

$$f(x^*) + g(x^*)u^* = 0 \tag{2}$$

Under this condition, $x(t) = x^*$ is a constant solution of the system. This definition characterizes the equilibrium of the system under a fixed control law.

The equilibrium point $x^*$ is locally stable if, for every $\varepsilon > 0$, there exists a $\delta > 0$ such that:

$$\|x(t_0) - x^*\| < \delta \;\Rightarrow\; \|x(t) - x^*\| < \varepsilon, \quad \forall\, t \geq t_0 \tag{3}$$

for any initial time $t_0 \geq 0$. This implies that trajectories starting close to $x^*$ remain in its neighborhood for all future time.

The equilibrium point $x^*$ is locally asymptotically stable if it is locally stable and, in addition,

$$\lim_{t\to\infty} x(t) = x^* \tag{4}$$

for all initial conditions $x(t_0)$ sufficiently close to $x^*$.

These notions form the basis for Lyapunov-based stability analysis of nonlinear systems with inputs [19] [20].

### B. Lyapunov Stability

Under a fixed state-feedback control law $u = \pi(x)$, system (1) reduces to an autonomous closed-loop system with equilibrium point $x^*$ satisfying (2). A continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}$ is said to be a Lyapunov function candidate if:

$$V(x^*) = 0, \quad V(x) > 0 \;\; \forall\, x \neq x^* \tag{5}$$

The time derivative of $V(x)$ along the trajectories of the closed-loop system is given by:

$$\dot{V}(x) = \nabla V(x)^\top \left(f(x) + g(x)\pi(x)\right) \tag{6}$$

If $\dot{V}(x) \leq 0$ in a neighborhood of $x^*$, the equilibrium point $x^*$ is stable. If $\dot{V}(x) < 0$ for all $x \neq x^*$ in that neighborhood, the equilibrium point $x^*$ is locally asymptotically stable. The classical Lyapunov stability results used in this work are summarized in [20].

### C. Markov Decision Process and Reinforcement Learning

A reinforcement learning problem can be formulated as a *Markov Decision Process* (MDP), defined by the tuple:

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$$

where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ the action space, $\mathcal{P}(s' \mid s, a)$ the state transition probability, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function, and $\gamma \in (0, 1]$ the discount factor.

A policy $\pi : \mathcal{S} \to \mathcal{A}$ defines a mapping from states to actions. In the context of control systems, the system state $x(t)$ is associated with the MDP state $s_t$, while the control input $u(t)$ corresponds to the action $a_t$. At each time step, the DRL agent observes the current state, selects an action according to the policy, receives a scalar reward, and transitions to the next state according to the environment dynamics. The objective of DRL is to determine a policy that maximizes the expected cumulative reward over time. When a fixed policy $\pi$ is applied, the resulting closed-loop system becomes autonomous, allowing classical notions of equilibrium and Lyapunov stability to be applied. The MDP and DRL concepts used in this work follow standard definitions in the literature [21] [22]. Having established the system dynamics, equilibrium concepts, and Lyapunov stability foundations in this section, the integration of the Lyapunov function with the DRL framework is developed in the next section.

## III. Lyapunov Post-Shielding mechanism

Considering a controlled dynamical system with state $s \in \mathbb{R}^n$ and control input $u \in \mathbb{R}^m$, we propose a Lyapunov post-shielding mechanism that enforces stability-like behavior for a DRL learned controller without modifying the training process. Although the underlying controlled system may be continuous-time, the DRL controller is usually executed at discrete sampling instants and the selected action is held

constant over each control interval. The closed-loop behavior at the sampling instants is conveniently represented by an equivalent discrete-time system:

$$s_{k+1} = F(s_k, u_k, w_k) \tag{7}$$

where $k \in \mathbb{Z}$ denotes the control step, $F(\cdot)$ is the one-step transition map induced by the controlled plant and the numerical integration over one sampling period, and $w_k$ represents external effects such as modeling error or bounded actuation disturbance. The learned policy produces a nominal action:

$$u_k^{\text{nom}} = \pi_\theta(s_k) \tag{8}$$

where $s_k$ is the state available to the policy, and $\theta$ denotes the policy parameters. The key idea of post-shielding is to compute, at runtime and at each step, a minimally modified action $u_k$ that is as close as possible to $u_k^{\text{nom}}$ while guaranteeing a Lyapunov decrease condition (or a relaxed variant) for a chosen Lyapunov candidate $V : \mathbb{R}^n \to \mathbb{R}$.

Let $V(s)$ be a positive definite candidate according to (5) that measures deviation from the desired equilibrium point $x^*$ (typically the origin). In discrete time, a standard sufficient condition for stability-like behavior is that $V$ decreases along closed-loop trajectories. Strict decrease of the form $V(s_{k+1}) - V(s_k) \leq -\alpha(\|s_k\|)$ may be overly restrictive in real-world learning control setups where sampling, noise, and model mismatch are expected. In our proposed method, we adopt a relaxed Lyapunov contractive constraint of the form:

$$V(s_{k+1}) \leq (1+\rho)\, V(s_k) + c \tag{9}$$

with design parameters $\rho \geq 0$, which introduces a *multiplicative* slack (allowing a small proportional increase), and $c \geq 0$, which introduces an *additive* slack (allowing a fixed offset). When $\rho = 0$ and $c = 0$, the condition reduces to enforcing a non-increasing Lyapunov value. For $\rho > 0$ and/or $c > 0$, (9) permits bounded transient growth and helps accommodate discretization effects and external disturbances, while still guiding trajectories toward the equilibrium region. Define the one-step Lyapunov residual as

$$g_L(s_k, u_k, w_k) \triangleq V\big(F(s_k, u_k, w_k)\big) - (1+\rho)\, V(s_k) - c \tag{10}$$

where $g_L(\cdot)$ quantifies the extent to which the Lyapunov decrease condition is violated at time step $k$ (with $g_L > 0$ indicating a violation and $g_L \leq 0$ indicating satisfaction). The objective of post-shielding is to select a corrected action $u_k$ that enforces $g_L(s_k, u_k, w_k) \leq 0$ while remaining as close as possible to the nominal action $u_k^{\text{nom}}$ and respecting the actuator bounds. In many applications, the control input is constrained to a compact accepted set $\mathcal{U} \subset \mathbb{R}^m$. In DRL-based control, a common choice is a componentwise box constraint:

$$\mathcal{U} = \{u \in \mathbb{R}^m : \; u_{\min} \leq u \leq u_{\max}\} \tag{11}$$

At each time step, the post-shield computes a minimally modified action by solving a local projection problem:

$$u_k \in \arg\min_{u \in \mathcal{U}} \; \|u - u_k^{\text{nom}}\|_2^2 \quad \text{s.t.} \quad g_L(s_k, u, w_k) \leq 0 \tag{12}$$

thereby enforcing the Lyapunov-based condition in (9) while otherwise preserving the behavior suggested by the learned policy. The constraint in (12) is generally nonlinear, since both the one-step map $F$ and the Lyapunov candidate $V$ may be nonlinear. To obtain a computationally cheap online mechanism, the nonlinear constraint is enforced using a first-order (local) approximation around the nominal action.

$$\begin{aligned} g_{L0} &\triangleq g_L(s_k, u_k^{\text{nom}}, w_k)\,, \\ \nabla_u g_{L0} &\triangleq \left.\frac{\partial g_L(s_k, u, w_k)}{\partial u}\right|_{u = u_k^{\text{nom}}} \end{aligned} \tag{13}$$

where $\nabla_u g_0$ can be approximated via finite differences using a one-step state predictor. Using the first-order Taylor expansion of $g_L$ with respect to $u$ yields the local surrogate constraint:

$$g_L(s_k, u, w_k) \approx g_{L0} + \nabla_u g_{L0}^\top (u - u_k^{\text{nom}}) \leq 0 \tag{14}$$

which provides a tractable sufficient condition in a neighborhood of $u_k^{\text{nom}}$. Substituting (14) into (12) yields a small Quadratic Program (QP) with a single linear inequality and box bounds:

$$\begin{aligned} u_k \in \arg\min_{u \in \mathcal{U}} \; & (u - u_k^{\text{nom}})^\top (u - u_k^{\text{nom}}) \\ & \text{where } \nabla_u g_{L0}^\top u \leq \nabla_u g_{L0}^\top u_k^{\text{nom}} - g_{L0} \end{aligned} \tag{15}$$

If $g_{L0} \leq 0$, the nominal action already satisfies the Lyapunov condition and the shield remains inactive, *i.e.*, $u_k = u_k^{\text{nom}}$. If $g_{L0} > 0$, the shield computes the closest permissible action that satisfies the half-space constraint in (15).

A critical aspect is robustness to bounded actuation disturbances and modeling errors. Suppose the actual applied action differs from the commanded action due to an unknown but bounded disturbance $\varepsilon_k$, such that:

$$\begin{aligned} u_k^{\text{app}} &= \text{clip}(u_k + \varepsilon_k,\; u_{\min},\; u_{\max})\,, \\ \varepsilon_k &\in \mathcal{E} \triangleq \{\varepsilon \in \mathbb{R}^m : \; \|\varepsilon\|_\infty \leq \varepsilon_{\max}\} \end{aligned} \tag{16}$$

where $\text{clip}(v,\, v_{\min},\, v_{\max}) \triangleq \max(v_{\min},\, \min(v_{\max},\, v))$ denotes the elementwise saturation operator. To avoid underestimating the disturbance, the shield adopts a conservative envelope $\varepsilon_{\max}$ that upper-bounds the disturbance magnitude. The Lyapunov condition is then enforced in a worst-case sense:

$$\begin{aligned} \max_{\varepsilon \in \mathcal{E}} \Big[ & V\big(F(s_k,\; \text{clip}(u + \varepsilon,\; u_{\min},\; u_{\max}),\; w_k)\big) \\ & - (1+\rho)\, V(s_k) - c \Big] \leq 0 \end{aligned} \tag{17}$$

Accordingly, define the robust one-step residual:

$$\begin{aligned} g_L^{\text{rob}}(s_k, u) \triangleq \max_{\varepsilon \in \mathcal{E}} \Big[ & V\big(F(s_k,\; \text{clip}(u + \varepsilon,\; u_{\min},\; u_{\max}),\; w_k)\big) \\ & - (1+\rho)\, V(s_k) - c \Big] \end{aligned} \tag{18}$$

The post-shielding step enforces robustness by replacing $g_L$ in (12)–(15) with the worst-case residual $g_L^{\text{rob}}$, *i.e.*, it selects $u_k$ such that $g_L^{\text{rob}}(s_k, u_k) \leq 0$. Since the maximization in (18) is costly online, it is approximated by

evaluating a small set of extreme disturbances (*e.g.*, $\varepsilon \in \{-\varepsilon_{\max}, 0, +\varepsilon_{\max}\}$).

In addition to the Lyapunov decrease constraint, we enforce an energy bound to prevent excessively energetic motions and improve robustness under actuation disturbances. Let $E : \mathbb{R}^n \to \mathbb{R}$ denote a nonnegative energy proxy, and let $E_{\max} > 0$ be a prescribed limit. The energy constraint is imposed as a one-step condition:

$$E(s_{k+1}) \le E_{\max} \tag{19}$$

which can be written via the residual

$$g_E(s_k, u, w_k) \triangleq E(F(s_k, u, w_k)) - E_{\max} \tag{20}$$

Overall, Lyapunov post-shielding can be viewed as a runtime safety/stability filter placed after the policy output. It leaves the learning algorithm unchanged and introduces a lightweight optimization layer that (i) checks whether the nominal action satisfies a Lyapunov-type one-step condition and an energy bound on the predicted next state, (ii) if either condition is violated, projects the action to the nearest feasible input under actuator bounds, and (iii) enforces a robust version of these constraints under bounded action disturbances by evaluating worst-case residuals over a disturbance set. This design provides a practical mechanism to improve closed-loop constraint compliance under the tested disturbances, while preserving the nominal policy behavior whenever possible.

## IV. Results and Discussion

To simultaneously enforce the Lyapunov and energy conditions, we combine the residual in (10) and (20) as:

$$g_{\max}(s_k, u, w_k) \triangleq \max\big(g_L(s_k, u, w_k),\, g_E(s_k, u, w_k)\big) \tag{21}$$

We then require $g_{\max}(s_k, u, w_k) \le 0$, which is equivalent to jointly enforcing $g_L(s_k, u, w_k) \le 0$ and $g_E(s_k, u, w_k) \le 0$. The post-shield solves the same projection problem in (12)–(15), replacing $g_L$ with $g_{\max}$. In practice, the one-step energy term $E(s_{k+1})$ is evaluated using a one-step predictor, yielding a lightweight online mechanism that unifies the Lyapunov and energy constraints into a single scalar inequality.

### A. Simulation setup

To demonstrate a proof-of-concept of the proposed stability-like mechanism, we consider the `InvertedPendulum-v5` environment from `Gymnasium` with the MuJoCo physics engine. This task is closely related to the classical cart–pole setting [23], while MuJoCo provides a higher-fidelity continuous-time simulator and a continuous control interface. The environment has a single continuous control input $u \in [-3, 3]$, representing the horizontal force applied to the cart. The state (observation) is four-dimensional, $s = [x, \theta, \dot{x}, \dot{\theta}]^\top$, where $x$ is the cart position, $\theta$ is the pole angle, and $\dot{x}$ and $\dot{\theta}$ denote the corresponding linear and angular velocities (with unbounded ranges). To increase the feasibility of initial conditions, we replace the default reset and sample with $x \in [-0.75, 0.75]$ m, $\theta \in [-10, 10]^\circ$, $\dot{x} \in [-0.1, 0.1]$ m/s, and $\dot{\theta} \in [-10, 10]^\circ$/s at the beginning of each episode. The environment reward follows the standard specification, a +1 per-step survival reward while the episode remains valid. The total energy of the system is computed as:

$$E(s) = \tfrac{1}{2} M \dot{x}^2 + mgL(1 - \cos\theta) + \tfrac{1}{2} m L^2 \dot{\theta}^2 \tag{22}$$

which combines the cart translational kinetic energy, the pendulum gravitational potential energy, and the pendulum rotational kinetic energy. $M$ is the cart mass, $m$ and $L$ are the pole mass and length, and $g$ is the gravitational acceleration.

We employ the Proximal Policy Optimization (PPO) algorithm using the `stable-baselines3` library. PPO is an on-policy actor–critic method with a stochastic policy $\pi_\theta(u \mid s)$ and a value function $V_\phi(s)$. For continuous actions, the policy is modeled as a diagonal Gaussian, and both actor and critic are implemented as Multilayer Perceptron (MLP). Table I summarizes the PPO hyperparameters used in our experiments.

TABLE I
PPO ALGORITHM HYPERPARAMETERS.

| **Hyperparameter** | **Value** |
|---|---|
| Learning rate | $1e{-4}$ |
| Number of steps ($n_{\text{steps}}$) | 4096 |
| Number of epochs K ($n_{\text{epochs}}$) | 10 |
| Batch size N | 512 |
| Clip range ($\epsilon$) | 0.2 |
| Discount factor ($\gamma$) | 0.99 |
| Generalized Advantage Estimation (GAE) parameter ($\lambda$) | 0.95 |
| Entropy coefficient ($c_{\text{ent}}$) | 0.01 |
| Max grad norm | 0.5 |
| Optimizer | Adam |

### B. Robust Constraint Satisfaction and State Regulation

After training the vanilla PPO agent on Inverted Pendulum environment without noise ($\sigma_{\text{train}} = 0$), we assess robustness under actuation uncertainty by injecting stochastic pre-step noise during evaluation ($\sigma_{\text{test}} = 0.1$). At each timestep $k$, the commanded action $u_{\text{cmd}}$ is perturbed before being applied to the environment according to:

$$u_{\text{applied}} = \text{clip}(u_{\text{cmd}} + \varepsilon_k,\ -3,\ 3)\,, \qquad \varepsilon_k \sim \mathcal{N}\big(0, \sigma^2_{\text{test}}\big) \tag{23}$$

where $\varepsilon_k$ is further clipped to $\varepsilon_k \in [-k_{\text{clip}}\sigma_{\text{test}},\, k_{\text{clip}}\sigma_{\text{test}}]$ with $k_{\text{clip}} = 1.5$. The policy parameters are frozen during evaluation (inference-only), and the disturbance is activated only within a fixed window $k \in [200, 800)$ in each episode (maximum horizon 1000 steps). The policy produces the nominal action $u_{\text{nom}}$, and the post-shield (if active) maps it to a constrained command $u_{\text{cmd}}$. Finally, after actuator disturbances are added, $u_{\text{applied}}$ is the applied action. To evaluate constraint satisfaction, we monitor the per-step Lyapunov residual $g_L$ defined in (10), evaluated under the applied action $u_k^{\text{applied}}$. We define the per-step margin $mk_k \triangleq g_L(s_k,\, u_k^{\text{applied}},\, w_k)$, which is negative when the Lyapunov decrease condition is satisfied and positive otherwise. We report three statistics: the violation rate `mk_viorate` (fraction of timesteps where $mk_k > \varepsilon_{\text{vio}}$, for a small threshold

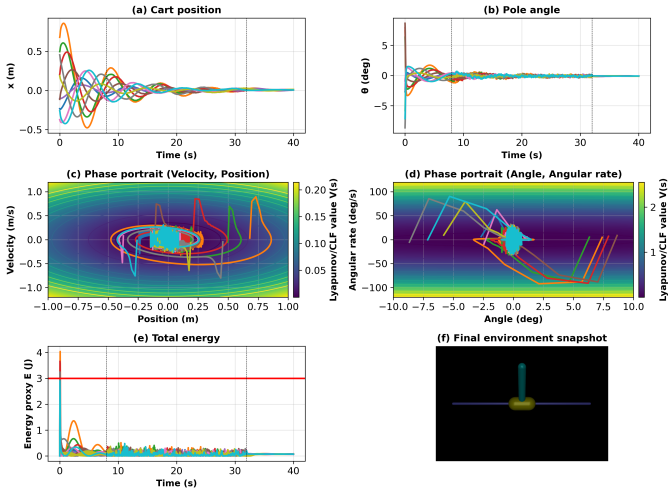


Fig. 1. Time-series and phase-portrait visualization for `InvertedPendulum-v5` using the vanilla PPO policy (no shield) under stochastic pre-step actuation noise.

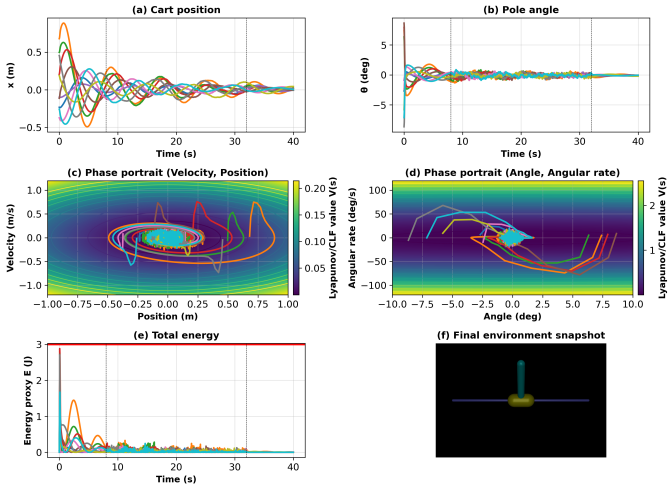


Fig. 2. Time-series and phase-portrait visualization for `InvertedPendulum-v5` using the vanilla PPO policy augmented with the proposed post-shield under stochastic pre-step actuation noise.

$\varepsilon_{\text{vio}} > 0$), the mean margin `mk_mean`, and the worst-case margin `mk_max`.

Figure 1 and Figure 2 provide a qualitative, trajectory-level comparison between the vanilla unshielded and post-shield PPO policies. Panels (a)–(b) show the cart position and pole angle responses across multiple rollouts. While both methods stabilize the pendulum and reach the maximum return, the post-shield policy shows larger short-term deviations after the disturbance starts, resulting in a wider spread of trajectories. The phase portraits in panels (c)–(d) further highlight the difference in closed-loop behavior. Without shielding, several trajectories undergo intermittent bursts that move away from the low-energy region around the origin, consistent with the non-negligible constraint violation frequency (`mk_viorate` $= 0.083 \pm 0.006$ and `mk_max` $= 0.844$). In contrast, post-shielding concentrates the trajectories closer to the equilibrium region and suppresses these

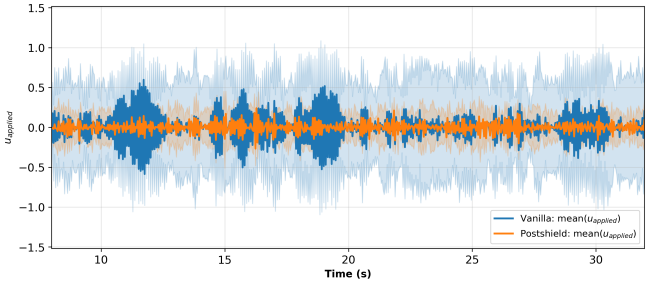


Fig. 3. Applied control input $u_{\text{applied}}$ for vanilla PPO and vanilla+post-shield under stochastic pre-step actuation noise. Solid lines denote the mean across rollouts and the shaded regions indicate variability.

excursions, aligning with the substantially improved MK statistics (`mk_viorate` $= 0.004 \pm 0.002$ and `mk_max` $= 0.190$). Panel (e) illustrates the energy proxy and the enforced bound $E_{\max}$. In the vanilla case, several rollouts exhibit pronounced energy spikes, occasionally exceeding the threshold, indicating high-energy transients, especially at the beginning of the rollouts. With post-shielding, the energy trajectories remain below the bound, confirming that the energy threshold constraint effectively limits one-step energy growth in the shield's next-state prediction. This provides an additional safeguard against high-energy transients (*e.g.*, sudden accelerations or large excursions) during the disturbance window. These safety gains are achieved with small online action corrections ( $|du|_{\text{mean}} = 3.52\text{e}{-2}$), even though the shield activates frequently (`shield_use`$= 0.614$) due to the long disturbance window and noise-free training. The dashed vertical lines indicate the start and end of the disturbance window. In particular, both methods achieve the maximum return (1000), indicating reward saturation and limited sensitivity to safety degradations. For this reason, the safety and stability discussion focuses on the constraint indicators (MK statistics). Overall, the results support the main claim that execution-time post-shielding can substantially improve robustness and constraint compliance under actuation noise while remaining close to the nominal policy. Figure 3 compares the applied control input (action) for the vanilla PPO policy and the same policy with post-shielding under stochastic pre-step actuation disturbances. During the shown noise interval, the vanilla policy shows larger fluctuations and higher dispersion in applied action across rollouts, reflecting stronger sensitivity to actuation disturbance. In contrast, the post-shield moderates the applied input and reduces its variability, consistent with the improved constraint satisfaction reported in the MK metrics.

### C. Results under Sweeping of the Disturbance Value

We evaluate robustness by sweeping the actuation disturbance level $\sigma_{\text{test}}$ while keeping the PPO policy fixed. Since the task return saturates across all disturbance levels, the comparison focuses on the Lyapunov-based constraint indicator (MK statistics). As the disturbance increases, the unshielded policy shows more frequent constraint violations and larger transient excursions. In contrast, the proposed

TABLE II

ROBUSTNESS SWEEP OVER ACTUATION DISTURBANCE $\sigma_{\text{test}}$.

| $\sigma$ | Meth. | mk_viorate | mk_mean | mk_max | shield_use | $\lvert du \rvert_{\text{mean}}$ |
|---|---|---|---|---|---|---|
| 0.05 | Van. | $0.014 \pm 0.003$ | $-1.94\text{e}{-2}$ | $8.41\text{e}{-1}$ | 0 | 0 |
| 0.05 | PS | $0.001 \pm 0.001$ | $-1.54\text{e}{-2}$ | $2.10\text{e}{-1}$ | 0.378 | $9.59\text{e}{-3}$ |
| **0.10** | Van. | $0.083 \pm 0.006$ | $-1.95\text{e}{-2}$ | $8.44\text{e}{-1}$ | 0 | 0 |
| **0.10** | PS | $0.004 \pm 0.002$ | $-1.35\text{e}{-2}$ | $1.90\text{e}{-1}$ | 0.614 | $3.52\text{e}{-2}$ |
| 0.15 | Van. | $0.131 \pm 0.007$ | $-2.05\text{e}{-2}$ | $8.49\text{e}{-1}$ | 0 | 0 |
| 0.15 | PS | $0.037 \pm 0.005$ | $-1.48\text{e}{-2}$ | $1.79\text{e}{-1}$ | 0.950 | $1.45\text{e}{-1}$ |
| 0.20 | Van. | $0.165 \pm 0.008$ | $-2.17\text{e}{-2}$ | $8.62\text{e}{-1}$ | 0 | 0 |
| 0.20 | PS | $0.091 \pm 0.008$ | $-1.57\text{e}{-2}$ | $1.86\text{e}{-1}$ | 0.965 | $2.32\text{e}{-1}$ |

*Van.* = vanilla (no shield), *PS* = post-shield.

post-shield consistently improves constraint satisfaction by suppressing worst-case violations and reducing the overall violation rate, at the cost of more frequent shield activation. The results in Table II indicates enhanced robustness under actuation noise while maintaining task performance.

### D. Optimization of the Lyapunov post-shield parameters

Since the shield behavior depends on the design parameters $\rho$, $c$, the Lyapunov weights $V_{\text{kwargs}} = \{k_{\theta_d}, k_x, k_{\dot{x}}\}$, and the robustness margin $k_{\text{robust}}$, these parameters are tuned using a random sweep. The robustness margin $k_{\text{robust}}$ scales the disturbance envelope in (18), setting $\varepsilon_{\max} = k_{\text{robust}} \cdot \sigma_{\text{test}}$. A larger $k_{\text{robust}}$ produces a more conservative shield that activates more frequently but reduces worst-case violations. We sample $N_{\text{sweep}}$ candidate configurations from predefined ranges and evaluate each on the fixed PPO policy under actuation disturbances. For each configuration, we compute the violation rate (mk_viorate), worst-case violation (mk_max), shield activation ratio, and mean action modification. Candidates are ranked using a weighted score that prioritizes minimizing mk_viorate while penalizing large worst-case violations and excessive intervention. All final results are computed on a disjoint set of test seeds, and the best configuration at $\sigma_{\text{test}} = 0.1$ is used for reporting.

## V. CONCLUSION AND FUTURE WORK

This paper presented a post-shield for improving robustness and constraint compliance of a pre-trained PPO policy under stochastic actuation disturbances present only at execution time. The proposed shield combines a Lyapunov constraint with a one-step energy bound, and applies a minimal action correction via a lightweight projection to satisfy the constraints. This mechanism is designed as a practical deployment-time safeguard rather than a full safety proof. Constraint satisfaction is enforced with respect to a one-step predictor and a bounded disturbance envelope; therefore, the results should be interpreted as empirical robustness improvements under the tested noise models. Overall, the results support the main claim that post-shielding provides a practical and modular mechanism to enforce stability and energy-related constraints for DRL controllers at deployment time without retraining. Future work will extend the approach to higher-dimensional industrial systems and multi-actuator settings, investigate theoretical guarantees under model mismatch and partial observability.

## REFERENCES


[1] B. Recht, "A tour of reinforcement learning: The view from continuous control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, no. 1, pp. 253–279, 2019.

[2] V. Gullapalli, *Reinforcement learning and its application to control*. University of Massachusetts Amherst, 1992.

[3] Z. Fan, W. Zhang, and W. Liu, "Multi-agent deep reinforcement learning-based distributed optimal generation control of dc microgrids," *IEEE Transactions on Smart Grid*, vol. 14, no. 5, pp. 3337–3351, 2023.

[4] R. Han, S. Chen, S. Wang, Z. Zhang, R. Gao, Q. Hao, and J. Pan, "Reinforcement learned distributed multi-robot navigation with reciprocal velocity obstacle shaped rewards," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 5896–5903, 2022.

[5] L. Yu, W. Xie, D. Xie, Y. Zou, D. Zhang, Z. Sun, L. Zhang, Y. Zhang, and T. Jiang, "Deep reinforcement learning for smart home energy management," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2751–2762, 2019.

[6] C. Pukdeboon, "A review of fundamentals of lyapunov theory," *J. Appl. Sci*, vol. 10, no. 2, pp. 55–61, 2011.

[7] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*, pp. 3420–3431, Ieee, 2019.

[8] H. Bevrani, M. R. Feizi, and S. Ataee, "Robust frequency control in an islanded microgrid: $H_\infty$ and $\mu$-synthesis approaches."," *IEEE transactions on smart grid*, vol. 7, no. 2, pp. 706–717, 2015.

[9] K. Kronander and A. Billard, "Passive interaction control with dynamical systems," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 106–113, 2015.

[10] M. G. Zarch, V. Puig, and J. Poshtan, "Verification of the control system performance using viability theory," in *2017 4th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 0778–0783, IEEE, 2017.

[11] T. Shafa, Y. Meng, and M. Ornik, "Reachable predictive control: A novel control algorithm for nonlinear systems with unknown dynamics and its practical applications," *arXiv preprint arXiv:2510.02623*, 2025.

[12] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[13] S. Huh and I. Yang, "Safe reinforcement learning for probabilistic reachability and safety specifications: A lyapunov-based approach," *arXiv preprint arXiv:2002.10126*, 2020.

[14] Z. Cao, R. Wang, X. Zhou, and Y. Wen, "Toward model-assisted safe reinforcement learning for data center cooling control: A lyapunov-based approach," in *Proceedings of the 14th ACM International Conference on Future Energy Systems*, pp. 333–346, 2023.

[15] T. Westenbroek, F. Castaneda, A. Agrawal, S. Sastry, and K. Sreenath, "Lyapunov design for robust and efficient robotic reinforcement learning," *arXiv preprint arXiv:2208.06721*, 2022.

[16] Y.-C. Chang and S. Gao, "Stabilizing neural control using self-learned almost lyapunov critics," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1803–1809, IEEE, 2021.

[17] H. I. Ugurlu, A. Redder, and E. Kayacan, "Lyapunov-inspired deep reinforcement learning for robot navigation in obstacle environments," in *2025 IEEE Symposium on Computational Intelligence on Engineering/Cyber Physical Systems (CIES)*, pp. 1–8, IEEE, 2025.

[18] K. Mizuta and K. Leung, "Cobl-diffusion: Diffusion-based conditional robot planning in dynamic environments using control barrier and lyapunov functions," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 13801–13808, IEEE, 2024.

[19] A. Isidori, *Nonlinear control systems: an introduction*. Springer, 1985.

[20] H. K. Khalil, *Nonlinear systems*, vol. 3. Prentice hall Upper Saddle River, NJ, 2002.

[21] R. S. Sutton, A. G. Barto, *et al.*, *Reinforcement learning: An introduction*, vol. 1. MIT press Cambridge, 1998.

[22] D. Bertsekas, *Reinforcement learning and optimal control*, vol. 1. Athena Scientific, 2019.

[23] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 5, pp. 834–846, 1983.