arXiv:2505.13549v1 [cs.RO] 19 May 2025

TD-GRPC: Temporal Difference Learning with Group Relative Policy Constraint for Humanoid Locomotion

Khang Nguyen[1], Khai Nguyen[2], An T. Le[3], Jan Peters[3,4,5], Manfred Huber[1], Vien Ngo[2], and Minh Nhat Vu[6]

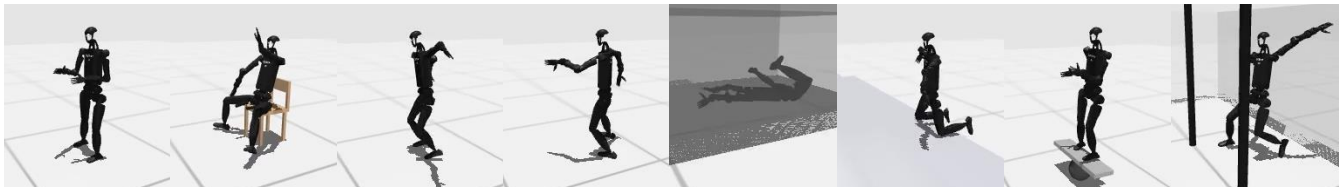

Fig. 1: Locomotion Tasks Performed by the Unitree H1-2 Humanoid with TD-GRPC: standing, sitting on a chair, walking, running, crawling under a tunnel, climbing stairs, balancing on a ball-board toy, and navigating through standing poles while avoiding collision.

***Abstract*— Robot learning in high-dimensional control settings, such as humanoid locomotion, presents persistent challenges for reinforcement learning (RL) algorithms due to unstable dynamics, complex contact interactions, and sensitivity to distributional shifts during training. Model-based methods, *e.g.*, Temporal-Difference Model Predictive Control (TD-MPC), have demonstrated promising results by combining short-horizon planning with value-based learning, enabling efficient solutions for basic locomotion tasks. However, these approaches remain ineffective in addressing policy mismatch and instability introduced by off-policy updates. Thus, in this work, we introduce Temporal-Difference Group Relative Policy Constraint (TD-GRPC), an extension of the TD-MPC framework that unifies Group Relative Policy Optimization (GRPO) with explicit Policy Constraints (PC). TD-GRPC applies a trust-region constraint in the latent policy space to maintain consistency between the planning priors and learned rollouts, while leveraging group-relative ranking to assess and preserve the physical feasibility of candidate trajectories. Unlike prior methods, TD-GRPC achieves robust motions without modifying the underlying planner, enabling flexible planning and policy learning. We validate our method across a locomotion task suite ranging from basic walking to highly dynamic movements on the 26-DoF Unitree H1-2 humanoid robot. Through simulation results, TD-GRPC demonstrates its improvements in stability and policy robustness with sampling efficiency while training for complex humanoid control tasks.**

I. Introduction

Humanoid locomotion is one of the most challenging domains in control and reinforcement learning (RL), due to its inherently high-dimensional and dynamically complex body structure [1], [2]. Achieving robust and adaptive control to accomplish locomotion tasks requires an algorithm that can not only plan effectively in the face of environmental uncertainty but also learn generalizable behaviors from limited iterations. While model predictive control (MPC) offers strong short-term decision-making capabilities via real-time trajectory optimization, its effectiveness is often bounded by model accuracy alongside the need for meticulously handcrafted cost functions. In contrast, RL enables more flexible, data-driven behavior learning but typically suffers from sampling inefficiency and unstable policy updates, especially for model-based RL (MBRL) approaches [3]–[5].

While MBRL holds great promises for sample efficiency by leveraging learned dynamics models for planning and value estimation [13], [14], it often suffers from compounding model errors and planning biases in games [8], [9], [15], [16], in quadruped robots control [17]–[19], and particularly in high-dimensional tasks such as in humanoid control [12]. Offline approaches to MBRL have attempted to mitigate these limitations by decoupling data collection from learning [5], [14]. Additionally, PlaNet [20] and Dreamer variants [20]–[22] have explored latent dynamics for improving policy learning from pixels [20], [23]. Nevertheless, many of these methods still exhibit weak generalization and lack the temporal abstraction necessary for robust and trustworthy long-horizon reasoning [4], [24], [25].

Earlier works have attempted to merge planning and learning more tightly. Temporal-Difference Model Predictive Control (TD-MPC) [26] addresses these issues by training latent dynamics models jointly with value functions using TD learning, avoiding the pitfalls of purely supervised model learning and enabling better credit assignment over long horizons. TD-MPC2 [11] further refines this framework by introducing stabilized actor-critic updates in the latent space, achieving improved consistency in off-policy learning. However, challenges persist despite these improvements due to the mismatch between training targets and planner-induced policies, destabilizing off-policy MBRL [24], [27]. Moreover, the lack of constraints during policy improvement steps can result in aggressive updates that cause significant distribution shifts, as studied in [28]–[31].

To bridge this gap, recent advancements have demonstrated promising results by combining the planning effi-

[1]University of Texas at Arlington, Texas, USA
[2]VinRobotics, Hanoi, Vietnam
[3]Intelligent Autonomous Systems Lab, TU Darmstadt, Germany
[4]German Research Center for AI (DFKI), SAIROL, Darmstadt, Germany
[5]Hessian.AI, Darmstadt, Germany
[6]Automation & Control Institute (ACIN), TU Wien, Vienna, Austria
E-mails: khang.nguyen8@mavs.uta.edu, minh.vu@ait.ac.at.

TABLE I: Comparison of policy learning- and optimization-based attributes across prior model-free and model-based approaches, including MPC [6], SAC [7], MuZero [8], EfficientZero [9], LOOP [10], TD-MPC2 [11], TD-M(PC)^2 [12], and our proposed method. The highlighted columns introduce policy constraints and group-relative rankings, which are critical for further policy optimization of humanoid locomotion.

Method \ Attr.	Continuous controller	Model learning objective	Value function learning	Policy constraint	Group-relative advantage	Inference strategy	Computation cost
MPC [6]	✓	not defined	✗	✗	✗	CEM	High
SAC [7]	✓	not defined	✓	✗	✗	policy	Low
MuZero [8]	✗	reward + value	✓	✗	✗	MCTS-based policy	Moderate
EfficientZero [9]	✗	reward + value	✓	✗	✗	MCTS-based policy	Moderate
LOOP [10]	✓	state	✓	✗	✗	CEM-based policy	Moderate
TD-MPC2 [11]	✓	reward + value + state	✓	✗	✗	CEM-based policy	Low
TD-M(PC)^2 [12]	✓	reward + value + state	✓	✓	✗	CEM-based policy	Low
TD-GRPC (Ours)	✓	reward + value + state	✓	✓	✓	CEM-based policy	Low

ciency of MPC with the adaptability of TD learning. Their variants extend this hybrid paradigm by incorporating latent dynamics models and stabilized value estimation, enabling more scalable and robust control in continuous settings. However, key challenges remain unresolved, particularly the mismatch between learned policy rollouts and the TD targets used in off-policy training, leading to degraded performance and instability over time and failing to accomplish the tasks.

In this work, we propose to alleviate these limitations by extending the TD-MPC framework [11] with Group Relative Policy Optimization (GRPO) [32] to improve stability and sample efficiency in humanoid locomotion settings. Inspired by recent state-of-the-art techniques in constrained policy optimization, our method further combines a trust-region constraint directly on the policy prior in the latent space, preserving planning feasibility while mitigating distributional drift during learning. In addition, our technique achieves a faster convergence rate than other baselines. Our contributions are summarized as follows:

- We propose a TD-GRPC framework, which integrates GRPO with explicit trust-region constraints in the latent space. In addition, we provide theoretical insights into its supporting role in stabilizing off-policy MBRL.
- We empirically validate TD-GRPC on the locomotion suite of `HumanoidBench` [33], demonstrating significant improvements in sample efficiency and policy robustness across a set of humanoid locomotion tasks.

II. Related Work

Temporal-Difference Model Predictive Control: Humanoid locomotion poses unique challenges in control due to its high-dimensional continuous action spaces, unstable dynamics, and complex contact behaviors from the environment [34]. TD-MPC has demonstrated its potential to address these difficulties by combining the short-horizon planning strategies of MPC-based approaches alongside RL's sample-efficient, value-driven adaptability. Through this line of research, notable works [10], [11], [26] have revealed that incorporating TD learning into MPC-based methods allows for learning flexible value functions without the need for meticulously designed cost functions. In particular, TD-MPC2 [11] extends the vanilla TD-MPC framework [26] by learning more scalable latent world models for continuous control, mitigating compounding model errors, and stabilizing planning. These insights are critical for humanoid locomotion, where model inaccuracies can rapidly destabilize gaits. Thus, by integrating TD learning with MPC-styled control paradigms, modern frameworks enhance sample efficiency, robustness to modeling errors, and adaptability to high-dimensional motor tasks, making them well-suited for complex humanoid behaviors. Our work further extends by algorithmically improving stability and data efficiency through GRPO, similar to language models, and with an additional explicit trust-region constraint for rolled-out policies from the latent representations.

Policy Constraint for Off-Policy Learning: Policy mismatching for robot learning, especially for humanoid locomotion, remains a fundamental challenge in off-policy learning, which is highly exposed to shifts between the learned rolled-out actions and TD targets. Moreover, the accumulation and compound of bootstrapping errors further induce cumulative errors and poor generalization [35], [36]. Offline-policy RL methods have greatly improved policy learning from fixed datasets. Previous works in this scope handled the distributional shifts by explicitly regularizing the learned policies towards the expert policies [35], [37]. In contrast, others [28], [38] employ importance sampling techniques to correct for out-of-distribution queries. Alternative approaches, such as in-sample learning [29], [39], recover reliable policies implicitly by restricting updates to observed actions, bypassing explicit distribution constraints. The off-policy challenge is also critical in MBRL when leveraging policy or value priors for planning. For instance, LOOP [10] proposes actor regularization control, introducing a conservatism mechanism during planning to stabilize online learning. Unlike these methods, our approach enforces distributional constraints directly on the policy prior without altering the planner, enabling greater flexibility in planning while preserving stability and fast policy updates.

To recap, we compare attributes between our proposed method and prior model-free and model-based approaches in Tab. I on policy constraint and optimization alongside model

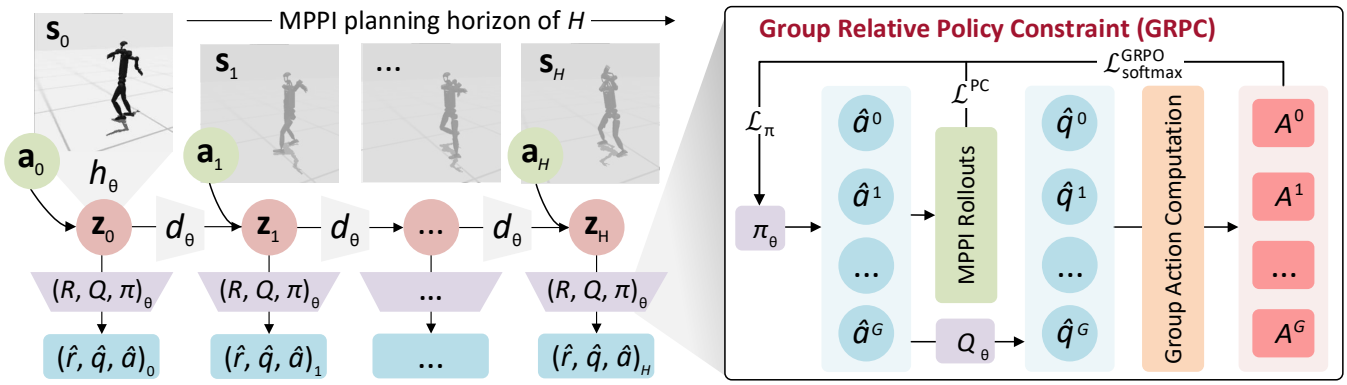

Fig. 2: Overview of TD-GRPC for Humanoid Locomotion: Starting from an initial state $\mathbf{s}_0$ encoded into latent state $\mathbf{z}_0$ with an encoder h_θ, a latent dynamics model d_θ takes an action $\mathbf{a}_t$ and the latent state $\mathbf{z}_t$ to predict the next latent state $\mathbf{z}_{t+1}$ across H steps of MPPI planning horizon. In each step, the reward, Q-value, and action are estimated via the MLPs R_θ, Q_θ, and π_θ, enabling latent-space planning with TD targets. At each state, sampled groups of actions are rolled out to evaluate Q-values, which are used to compute softmax-based advantage scores A^g of the g^{th} group. These scores are then used in the GRPC objective to guide the policy toward high-value actions while minimizing variance. To prevent excessive policy shifts, a trust-region constraint is imposed via a KL divergence penalty between the current and a prior MPPI-derived policies, enforcing residual learning with bounded policy divergence.

learning objectives, value function learning, and use cases.

III. Temporal Difference Learning with Group Relative Policy Constraint

A. Problem Formulation

Any humanoid locomotion task can be modeled as an infinite-horizon Markov Decision Process (MDP) characterized as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$, with $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the dynamics function, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in (0, 1]$ is the discount factor. The problem is to learn the parameters θ for the policy network $\Pi_\theta : \mathcal{S} \to \mathcal{A}$ that continuously controls the robot with optimal state-based actions by maximizing the discounted cumulative rewards along a trajectory Γ:

$$J^\pi = \mathbb{E}_{\Gamma \sim \Pi_\theta} \left[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t) \right], \tag{1}$$

where $\Gamma = (\mathbf{s}_t, \mathbf{a}_t)$ with each action $\mathbf{a}_t$ is sampled from the policy network $\Pi_\theta(\mathbf{s}_t)$, and $\mathbf{s}_t = \mathcal{P}(\mathbf{s}_{t-1}, \mathbf{a}_{t-1})$ is the transitional state based on the previous state and action.

Traditional model-free methods primarily focus on learning this policy directly, but they usually require abundant training data. On the other hand, model-based methods can be more sample-efficient as a learned dynamics model is used to simulate outcomes. Still, they often struggle with planning over long horizons due to inaccuracies in the model and high computation costs. Due to these, Hasen *et al.* [11], [26] proposed the integration of a sampling-based MPC method, such as MPPI [40], as a local trajectory optimizer for short-horizon planning using a learned latent dynamics model, and then extended it with a value function that predicts future returns. Action sequences of length H are sampled as latent trajectories generated by the learned dynamics model, and estimate the total return ϕ_Γ of a sampled trajectory Γ:

$$\phi_\Gamma = \mathbb{E}_\Gamma \left[\gamma^H Q_\theta(\mathbf{z}_H, \mathbf{a}_H) + \sum_{t=0}^{H-1} \gamma^t R_\theta(\mathbf{z}_t, \mathbf{a}_t) \right], \tag{2}$$

where $\mathbf{z}_t = h_\theta(\mathbf{s}_t)$ is the latent representation that selectively captures the relevant dynamics of the state $\mathbf{s}_t$, rather than all observation dimensions, $\mathbf{z}_t = d_\theta(\mathbf{s}_{t-1}, \mathbf{a}_{t-1})$ represents the next latent representation under the latent dynamics model d_θ, $\hat{r}_t = R_\theta(\mathbf{s}_t, \mathbf{a}_t)$ and $\hat{q}_t = Q_\theta(\mathbf{s}_t, \mathbf{a}_t)$ denote the predicted reward and value under MLPs with $\mathbf{a}_t \sim \mathcal{N}(\mu_t, \sigma_t)$ describes the trajectory distributions in Eq. 2, which are expressed as:

$$\mu_t = \frac{\sum_{i=1}^{k} \Omega_i \Gamma_i^*}{\sum_{i=1}^{k} \Omega_i}, \quad \sigma_t^2 = \frac{\sum_{i=1}^{k} \Omega_i (\Gamma_i^* - \mu^j)^2}{\sum_{i=1}^{k} \Omega_i}, \tag{3}$$

where $\Omega_i = \exp(\tau \phi^*_{\Gamma,i})$, τ is a temperature parameter, and Γ_i^* denotes the i^{th} of top-k trajectory corresponding to return estimate ϕ_Γ^*. In the RL-based robot learning context, Eq. 2 can therefore be called as an H-step look-ahead policy, where the RL-based planner iteratively maximizes the cost (reward) of the first step of the planning horizon until the learning objective for locomotion is accomplished.

B. TD Learning with Group Relative Policy Constraint

1) Policy Constraint as Residual Learning: In TD-MPC2 [11], the value function is learned using approximate policy iteration, similar to conventional off-policy RL methods. To further understand how various sources of approximation error influence the overall performance of the H-step look-ahead planner policy, we look into Lemma 3.1 from [41], which characterizes the asymptotic sub-optimality of the planner-induced policy μ_k in terms of the value approximation error of the policy π_k, the model prediction error in total variation distance, and the planner sub-optimality.

Lemma 3.1 (Singh and Yee [41]): Suppose we have a value function $V^\pi(\mathbf{s}) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t) \mid \mathbf{s}_0 = \mathbf{s} \right]$ such that $\max_s |V^*(\mathbf{s}) - \hat{V}(\mathbf{s})| \leq \varepsilon_v$. The performance of the 1-step greedy policy $\pi_{\hat{V}}$ is bounded by:

$$\frac{1-\gamma}{2\gamma} \left| J^{\pi^*} - J^{\pi_{\hat{V}}} \right| \leq \varepsilon_v$$

Following Theorem 1 in [10], we assume that the nominal policy π_k is obtained through approximate policy iteration,

Algorithm 1: TD-GRPC

Input : T : trajectory length, H: planning horizon of MPPI, G: number of groups, S: number of iterations, $\mathcal{D}$: latent buffer

function `td_grpc_training`$(T,\ H,\ G,\ S,\ \mathcal{D})$

  while *training* **do**

    // collecting trajs using latent-space MPPI

    for $t = 0, \ldots, T$ **do**

      $\mathbf{a}_t \sim \Pi_\theta(h_\theta(\mathbf{s}_t))$

      $(\mathbf{s}_{t+1}, r_t) \sim \mathcal{P}(\mathbf{s}_t, \mathbf{a}_t), \mathcal{R}_\theta(\mathbf{s}_t, \mathbf{a}_t)$

      $\mathcal{D} \leftarrow \mathcal{D} \cup (\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$

    for *step = 0, …, S* **do**

      // H-step sampling from latent buffer

      $\{\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1}\}^g_{t:t+H} \sim \mathcal{D}$ **for** G groups

      $\mu^G, \sigma^G =$ `compute_moments`$(\mathbf{a}_t)$ (Eq. 3)

      $\mathbf{z}_t = h_\theta(\mathbf{s}_t)$ **if** $\mathbf{s}_t$ is the first observation

      for $i = t, \ldots, t+H$ **do**

        $\mathbf{z}_{i+1} = d_\theta(\mathbf{z}_i, \mathbf{a}_i)$ (Eq. 10a)

        $\hat{r}_i = R_\theta(\mathbf{z}_i, \mathbf{a}_i)$ (Eq. 10b)

        // group sampling & policy constraint

        for $g = 1, \ldots, G$ **do**

          $\widehat{\mathbf{a}}^g_i \sim \pi_\theta(\mathbf{z}_i)$

          $\varepsilon = (\widehat{\mathbf{a}}^g_i - \mu^G)/\sigma^G$

          $\widehat{\mathbf{a}}^g_i \leftarrow$ `threshold`$(\widehat{\mathbf{a}}^g_i, \varepsilon)$ (Eq. 4)

          $\hat{q}^g_i = Q_\theta(\mathbf{z}_i, \widehat{\mathbf{a}}^g_i)$ (Eq. 10c)

        $A^g_i =$ `softmax`$(\hat{\mathbf{q}}^g)$ (Eq. 5)

        $\mathcal{L}^{(i)}_\pi = \frac{1}{G}\sum_{g=1}^{G} A^g_i \log \pi_\theta(\widehat{\mathbf{a}}^g_i \mid \mathbf{z}_i)$

      $\mathcal{L}_\pi = \frac{1}{H}\sum_{i=t}^{t+H}\left[\mathcal{L}^{(i)}_\pi + \beta\mathcal{L}_{\mathrm{KL}}\right]$ (Eq. 9)

      $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}_\pi$

and the resulting planner policy at the k^{th} iteration is denoted as $\mu_k = \pi_{H,k}$. Let the value approximation error be bounded as $\|Q_k - Q^{\pi_k}\|_\infty \leq \varepsilon_k$, the model approximation error be $\varepsilon_m = \max_{\mathbf{s},\mathbf{a}} \mathbf{D}_{\mathrm{TV}}(\mathcal{P} \mid\mid \hat{\mathcal{P}})$ with $\mathbf{D}_{\mathrm{TV}}$ as total variation distance, and the planner sub-optimality be $\varepsilon_{p,k}$. Let the reward function be bounded as $r(\mathbf{s},\mathbf{a}) \in [0, R_{\max}]$, and define the upper bound of the value function as $Q_{\max} = R_{\max}/(1-\gamma)$. Therefore, the planner's sub-optimal policy performance that satisfies the following uniform bound:

$$\limsup_{k\to\infty} \frac{1-\gamma^H}{2} |V^* - V^{\mu_k}| \leq$$

$$\limsup_{k\to\infty} \left[E(\varepsilon_m, H, \gamma) + \frac{\varepsilon_{p,k}}{2} + \frac{\gamma^H(1+\gamma^2)}{(1-\gamma)^2}\varepsilon_k\right],$$

where the model error constant E is defined as:

$$E(\varepsilon_m, H, \gamma) = R_{\max}\sum_{t=0}^{H-1}\gamma^t t\varepsilon_m + \gamma^H H\varepsilon_m V_{\max}.$$

Thus, with policies $\pi, \pi' \in \Pi$ and the reward's upper bound is $R_{\max}$, the policy divergence is lower bounded by the performance gap as the following expression:

$$\frac{(1-\gamma)^2}{2R_{\max}}\left|J^\pi - J^{\pi'}\right| \leq \max_{\mathbf{s}} \mathbf{D}_{\mathrm{TV}}(\pi' \mid\mid \pi)$$

To ensure stable policy updates in high-dimensional and sensitive control environments, we impose a trust-region constraint on the policy learning process with Kullback–Leibler (KL) divergence between roll-out and reference policies. Specifically, we constrain the updated policy π to remain close to a reference or prior policy π_{old} by imposing the following divergence condition $\mathbf{D}_{\mathrm{KL}}(\pi \mid\mid \pi_{\text{old}}) \leq \varepsilon$, where ε is a small adaptive threshold, enforcing a residual learning objective, encouraging the policy to improve upon the previous iteration while limiting drastic changes [42], [43]. Moreover, it mitigates the risk of destabilizing learned behaviors, critical in continuous control settings, especially for locomotion skills.

In practice, we implement this constraint as a policy constraint loss $\mathcal{L}^{\text{PC}}$ under the Lagrangian version [44]:

$$\mathcal{L}^{\text{PC}} = \max\left\{\mathbf{D}_{\mathrm{KL}}(\pi \mid\mid \pi_{\text{old}}) - \varepsilon,\ 0\right\}, \tag{4}$$

2) Group Relative Policy Constraint: We adopt and improve GRPO [32] to enhance action group-based explicit advantage refinement. Specifically, GRPO enhances entropy-regularized policy gradient methods by leveraging group-wise action comparisons, enabling the policy to learn from relative action preferences rather than relying on potentially noisy absolute value targets. In standard actor-critic methods, policy gradients are directly scaled by absolute Q-values or advantage estimates, which may be sensitive to reward scaling and value estimation errors. These limitations become especially pronounced in long-horizon tasks such as robotic locomotion. GRPO addresses this issue by constructing a relative preference distribution across sampled groups.

Formally, at each state $\mathbf{s}$, a set of G actions $\{\mathbf{a}_1, \ldots, \mathbf{a}_G\}$ is sampled, and their Q-values $\{q_1, \ldots, q_G\}$ are computed. These are used to define softmax-based advantage scores:

$$A_i(\mathbf{q}) = \frac{\exp(q_i/\tau)}{\sum_{j=1}^{G}\exp(q_j/\tau)}, \tag{5}$$

where $\mathbf{q} = Q_\theta(\mathbf{s}, \mathbf{a}_i)$ denotes the estimates, and τ is a temperature parameter and $0 \leq A_i(\cdot) \leq 1$ as its property.

As $\{\mathbf{a}_i\}_{i=1}^G$ is a group of G actions sampled from a policy $\pi_\theta(\mathbf{s})$ at the state $\mathbf{s}$ with $r_i = r_\theta(\mathbf{s}, \mathbf{a}_i)$ and $q_i = Q_\theta(\mathbf{s}, \mathbf{a}_i)$ are the reward and estimated value for each action, respectively, we assume that $||\nabla_\theta \log \pi_\theta(\mathbf{a} \mid \mathbf{s})|| = C$, and q_i, r_i are bounded above with $\forall \mathbf{a} \in \mathcal{A}$ and $\forall \mathbf{s} \in \mathcal{S}$. We obtain:

$$\mathrm{Var}\left[\nabla_\theta \mathcal{L}_{\text{softmax}}\right] \leq \mathrm{Var}\left[\nabla_\theta \mathcal{L}_{\text{std-norm}}\right], \tag{6}$$

with $\mathcal{L}_{\text{softmax}}$ and $\mathcal{L}_{\text{std-norm}}$ are the softmax-based and standard normalized advantage scores, respectively. Additionally, the normalized scores are unbounded, but the advantage scores are bounded in the range of 0 to 1, the variance of gradient of $\mathcal{L}_{\text{softmax}}$ is thus smaller than that of $\mathcal{L}_{\text{std-norm}}$:

$$||\nabla_\theta \mathcal{L}_{\text{softmax}}|| \text{ is bounded, } ||\nabla_\theta \mathcal{L}_{\text{std-norm}}|| \text{ is unbounded} \tag{7}$$

yields more stable policy updates at some constant C that asymptotically bounds $||\nabla_\theta \log \pi_\theta(\mathbf{a} \mid \mathbf{s})||$. Two keys favor softmax-based over normalized advantages. First, their outputs lie between 0 and 1, limiting the impact of outliers.

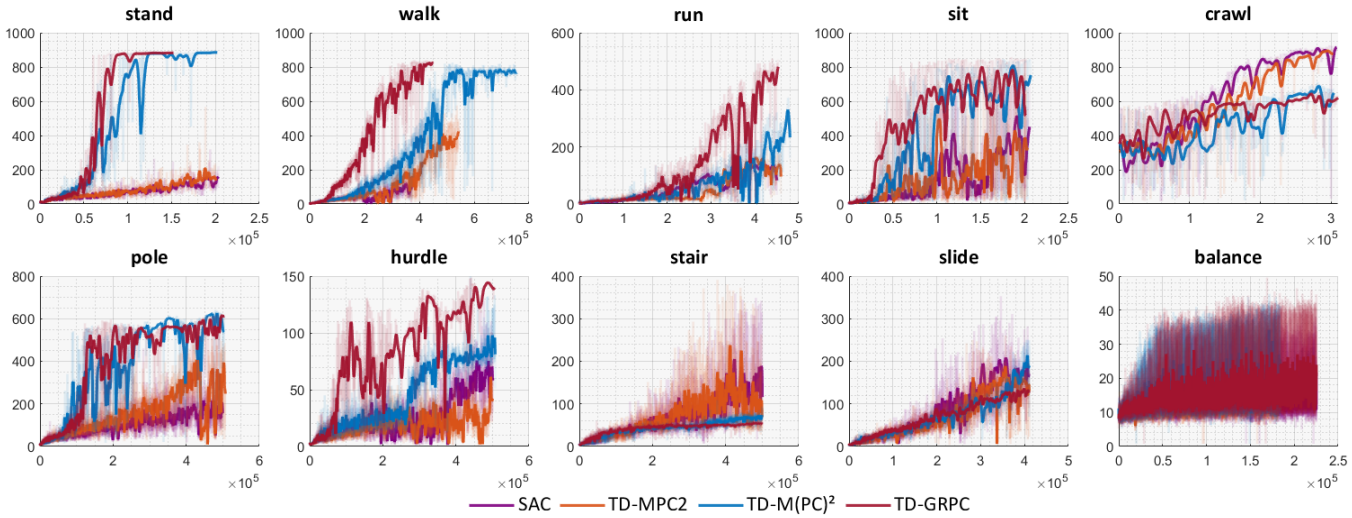

Fig. 3: Episode Returns of TD-GRPC and Baselines on H1–2 in Humanoid Locomotion Tasks: TD-GRPC achieves rapid convergence over others in standing, walking, running, sitting, navigating through poles, hurdling, and sliding tasks, while it performs worse in crawling tasks. In general, TD-GRPC shows slightly better data-efficiency than TD-M(PC)^2 and significantly better sampling-efficiency than SAC and TD-MPC2, alongside the fact that TD-GRPC outperforms TD-MPC2 and SAC on many tasks quantitatively. Nevertheless, all benchmarked algorithms fail to accomplish more challenging tasks, such as stair-climbing and balancing on a ball-board platform.

Meanwhile, normalized advantages induce large magnitudes under noise, leading to high-variance gradients. Second, policy gradients scale with the advantage values. If the advantage is very large or small, the gradient steps are disproportionately unstable. Therefore, softmax-based advantages smooth out extreme values and act like a soft attention mechanism, giving more stable updates.

With the group relative weights in Eq. 5 and based on Eq. 6 and Eq. 7, the improved GRPO objective is defined as:

$$\mathcal{L}_\pi^{\text{GRPO}}(\theta) = \frac{1}{G}\sum_{i=1}^{G} A_i(\mathbf{q}) \log \pi_\theta(\mathbf{a}_i \mid \mathbf{s}) \tag{8}$$

where μ_k denotes the behavior policy at k^{th} iteration from the buffer $\mathcal{D}$ obtained from Eq. 3. The KL constraint ensures the updated policy remains within a trust region of π. The overall policy objective combines Eq. 4 with Eq. 8:

$$\mathcal{L}_\pi(\theta) = \underbrace{\frac{1}{G}\sum_{i=1}^{G} A_i(\mathbf{q}) \log \pi_\theta(\mathbf{a}_i \mid \mathbf{s}))}_{\text{improved GRPO}} + \beta \underbrace{\log \mu(\mathbf{a} \mid \mathbf{s})}_{\text{policy constraint}}, \tag{9}$$

where β is a weighting coefficient controlling the penalty strength. The second term of Eq. 9 imposes a residual-style regularization equivalent to the trust-region [43].

Meanwhile, the latent dynamics d_θ, encoder h_θ, reward network R_θ, and value network Q_θ are concurrently optimized by the following model objective:

$$\mathcal{L}(\theta;\ \Gamma_i) = \|d_\theta(\mathbf{z}_i, \mathbf{a}_i) - h_\theta(\mathbf{s}_{i+1})\|_2^2 \tag{10a}$$
$$+ \|R_\theta(\mathbf{z}_i, \mathbf{a}_i) - r_i\|_2^2 \tag{10b}$$
$$+ \|Q_\theta(\mathbf{z}_i, \mathbf{a}_i) - [r_i + \gamma Q_\theta(\mathbf{z}_{i+1}, \pi_\theta(\mathbf{z}_{i+1}))]\|_2^2 \tag{10c}$$

The training procedure with temporal difference learning and group relative policy constraints is summarized in Alg. 1 along with the pipeline shown in Fig. 2. Meanwhile, the inference process remains the same as in TD-MPC [26] with cross-entropy method [45] on rolled-out learned policies.

IV. Experimental Results & Analysis

We evaluate our proposed method on `HumanoidBench` [33] with 10 locomotion tasks, including standing, walking, running, sitting on a chair, crawling under a tunnel, navigating through poles while avoiding collision, hurdling, climbing stairs, sliding, and balancing on a ball-board platform, on **the 26-DoF Unitree H1-2 humanoid** with two legs (6-DoF each $\times 2$) and two arms (7-DoF each $\times 2$), which is more dynamically flexible compared to H1 with 21 DoFs. While training, we also include the hands in the robot's model to ensure that the learned policies consider both hands' mass, although the robot does not encounter manipulation tasks. Additionally, H1-2 weighs about 70 kg compared to H1's 47 kg, presenting challenges for learning body dynamics due to its significantly heavier build. For comparison, we select the following three state-of-the-art RL methods as baselines:

- **Soft Actor-Critic (SAC)** [7]: a model-free, off-policy RL algorithm with maximum entropy RL [46].
- **TD-MPC2** [11]: the state-of-the-art MBRL that combines MPC with TD learning for diverse continuous control tasks in the DeepMind control suite [47].
- **TD-M(PC)^2** [12]: the most recent variant of TD-MPC2 for humanoid locomotion that utilizes KL-regularized policy learning to overcome value overestimation.

.

A. Quantitative Comparisons

As shown in Fig. 3, we compare the episode returns TD-GRPC against the baselines on the humanoid locomotion tasks mentioned. All algorithms are run with the planning horizon H of 3, size of latent buffer $\mathcal{D}$ of $1,000,000$, discount factor γ of 0.995, and learning rate of 0.0003 on an AMD Ryzen 9 7950X3D CPU and an NVIDIA RTX 4090 GPU. For TD-GRPC, we set the number of groups G as 3.

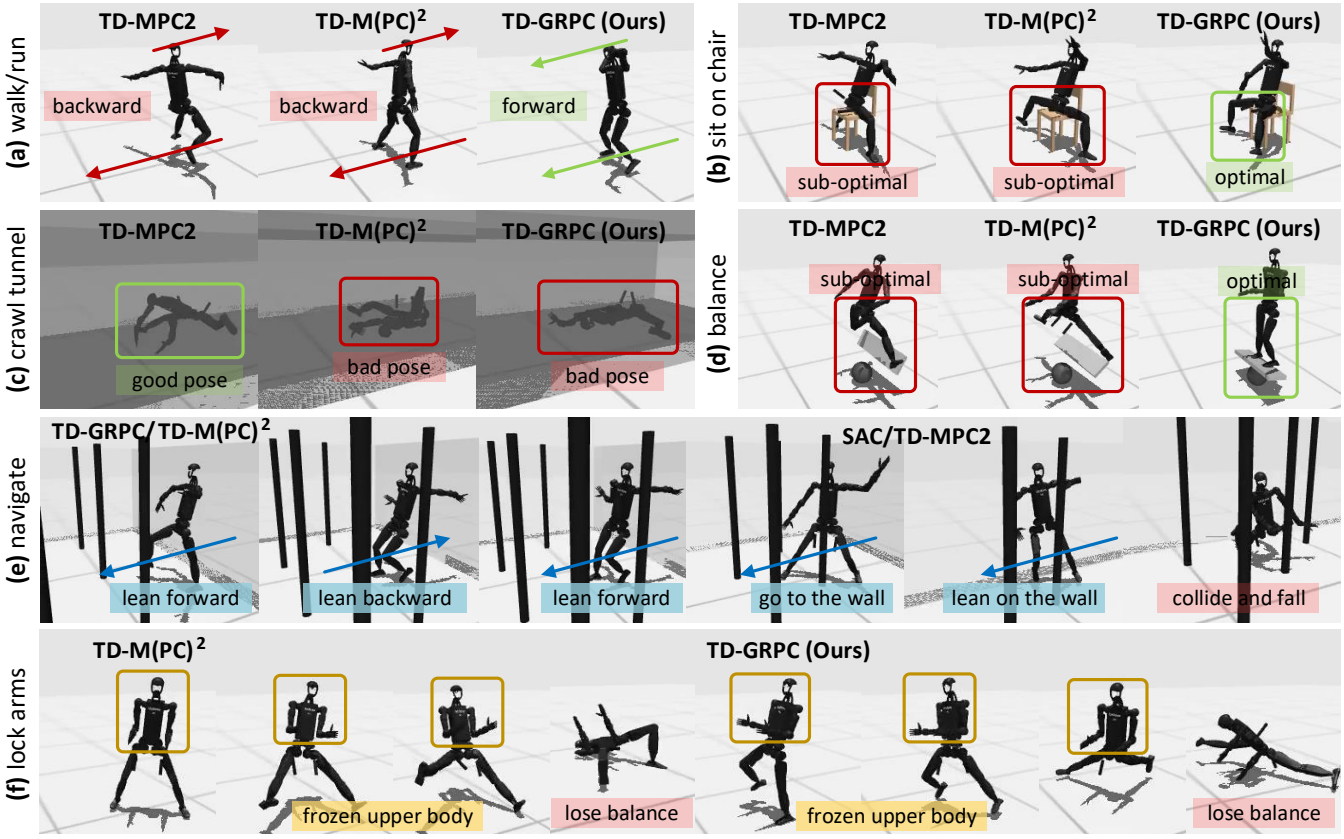

Fig. 4: Behavioral Analysis of H1-2 in Humanoid Locomotion Tasks: (a) *Walking and Running Backwards:* walking direction comparisons between TD-MPC2, TD-M(PC)^2, and our approach. TD-GRPC directs H1-2 to move forward, but TD-MPC2 and TD-M(PC)^2 make it walk/run backward. (b) *Sitting Stability:* sitting pose comparisons between TD-MPC2, TD-M(PC)^2, and TD-GRPC. TD-GRPC achieves optimal sitting leg pose. All frames are taken from the last state of the evaluation episode. (c) *Crawling Pose and Height:* The crawling pose and height produced by TD-MPC2 are better than those generated by TD-M(PC)^2 and TD-GRPC. (d) *Balancing is Hard for Heavy Body:* Due to its heavy body weight, all three methods suffer difficulties in keeping H1-2 balance itself on the platform. However, TD-GRPC can balance the robot for a short period; meanwhile, TD-MPC2 and TD-M(PC)^2 make the robot's legs flick the board away and fail immediately. (e) *Navigating through Standing Poles:* TD-M(PC)^2 and TD-GRPC induce a standing behavior without navigating, while SAC and TD-MPC2 produce valid motions but collide with poles and fail. (f) *Arm-Balancing Helps Locomotion:* Freezing the upper body of H1-2 makes walking and running unstable. This experimental finding is evaluated with both TD-M(PC)^2 and TD-GRPC.

1) Standing: TD-GRPC effectively trains H1-2 to stand upright and stably faster than TD-MPC2 and TD-M(PC)^2. Both TD-GRPC and TD-M(PC)^2 surpass the reward of 800. Meanwhile, SAC and TD-MPC2 fail to teach the robot to stand within $200,000$ iterations.

2) Walking: As with the standing task, H1-2, when trained with TD-GRPC, can walk more rapidly than other baselines. Both TD-M(PC)^2 and TD-GRPC gain the reward of more than 750 within $500,000$ iterations, but TD-GRPC allows the robot to walk at $400,000^{th}$ iteration. SAC and TD-MPC fail to enable the robot to walk during the same training period. However, the walking behavior among the algorithms is further analyzed in Sec. IV-B.1.

3) Running: The running task is defined analogously to the walking task, but with a goal speed higher than walking speed. While SAC and TD-MPC2 fail to generate proper running policies in $450,000$ iterations, TD-M(PC)^2 and TD-GRPC successfully enable the H1-2 to run, but TD-GRPC converges faster than TD-M(PC)^2 does. The illustrative behavior for this task is presented in Sec. IV-B.1.

4) Sitting: Similar to the standing and walking tasks, TD-GRPC continues showing faster convergence than SAC, TD-MPC2, and TD-M(PC)^2 in learning to sit on the chair. Unlike other algorithms, the experiments also confirm that the robot can sit steadily in a good pose with TD-GRPC, as described in Sec. IV-B.2. Both TD-M(PC)^2 and TD-GRPC-generated policies exceed the reward of 600 within $200,000$ iterations.

5) Crawling: SAC and TD-MPC2, on the other hand, show their superior capabilities to get the robot to crawl through the tunnel with the rewards of over 800 within $300,000$ iterations. TD-M(PC)^2 and TD-GRPC get the robot to crawl, but not enough to pass to the other end of the tunnel; their poses are visually explained in Sec. IV-B.3.

6) Balancing: The reward learned by TD-MPC2, TD-M(PC)^2, and TD-GRPC are approximately similar on the H1-2. The same phenomenon applies to SAC and TD-MPC2. This experiment is considered a hard task for H1-2 due to its heavy weight; the behavior is provided in Sec. IV-B.4.

7) Navigating through Standing Poles: The rewards gained by TD-M(PC)^2 and TD-GRPC are higher than those achieved by SAC and TD-MPC. Within $500,000$ train-

TABLE II: Solving ability of SAC [7], TD-MPC2 [11], TD-M(PC)^2 [12], and our proposed method, TD-GRPC, of locomotion tasks on H1-2 in `HumanoidBench` [33]: ✓ for tasks that are solved sufficiently, ● for tasks that need additional mild refinements for success, and ✗ for tasks that need further intensive policy-constraint learning of whole-body and selective environmental dynamics features.

Method \ Task	Locomotion Tasks in `HumanoidBench` Environment [33]									
	`stand`	`walk`	`run`	`sit`	`crawl`	`pole`	`hurdle`	`stair`	`slide`	`balance`
SAC [7]	✗	✗	✗	✗	✓	✗	✗	✗	●	✗
TD-MPC2 [11]	✗	✗	✗	●	✓	✗	✗	✗	●	✗
TD-M(PC)^2 [12]	✓	●	●	●	●	●	✗	✗	●	✗
TD-GRPC (Ours)	✓	✓	✓	✓	●	●	●	✗	●	●

ing iterations, TD-GRPC again shows its convergence well beyond TD-M(PC)^2, alluding that the task is concluded. Nevertheless, the robot's behaviors from TD-M(PC)^2 and TD-GRPC are distinct from those performed by SAC and TD-MPC2, as studied in Sec. IV-B.5.

8) Stair-Climbing: Learning whole-body dynamics is intricate for all baseline methods and our proposed method in stair-climbing tasks. Unlike walking or running, climbing stairs requires more than locomotion on an even surface; the robot has to change its foot height iteratively during stepping up. All methods struggle at a reward value of 200.

9) Hurdling: Hurdling is a running variant; however, the robot must jump over the tracks while running. During this task, TD-GRPC can let the robot jump over one track without collision, while others generate sub-optimal actions and fail. Evidently, TD-GRPC promotes the robot to learn as it produces higher learning reward curves than other methods, surpassing the reward threshold of 100.

10) Sliding: Similar to the stair-climbing objective, sliding requires the robot to climb up a hill-like landscape without needing foot-stepping behavior. All benchmarked methods and TD-GRPC are able to learn this task at the reward of 200, and generate physically-meaning actions (*i.e.*, knee-walking) to achieve the task's goal. However, the robot can only go up one hill and not go to other hills.

B. Qualitative Comparisons & Behavioral Analysis

Not just upon task completion, we outline selected humanoid locomotion tasks and analyze our findings on the robot's behavior as lessons learned for humanoid locomotion.

1) Walking & Running Backwards: While TD-MPC2 and TD-M(PC)^2 perform well on H1, they fail to correct walking and running poses on the H1-2, where they both make the robot walk or run backward with its head looking in the opposite direction. SAC is neither able to run nor walk the H1-2 properly. TD-GRPC successfully enables forward locomotion for walking and running, as shown in Fig. 4a.

2) Sitting Stability: The robot is able to sit stably on the chair with limited jerky motions while training with TD-GRPC, but not TD-MPC2 and TD-M(PC)^2. In addition, the leg poses are learned to be put appropriately when sitting, as shown in Fig. 4b. While trained with SAC and TD-MPC2, the robot can not sit stably on the chair.

3) Crawling Pose & Height: Righteously crawling with proper poses is more challenging for TD-M(PC)^2 and TD-GRPC. As shown in Fig. 4c, they both stuck at sub-optimal poses and could not crawl to the other end of the tunnel, failing to complete the tasks. On the contrary, TD-MPC2 completes its tasks with good body posture and head height.

4) Balancing is Hard for Heavy Body: Compared to H1, which is lighter, H1-2 suffers from different body dynamics to keep itself balanced on the ball-board platform. We find that TD-GRPC allows the robot to balance for a period of time before failing. In contrast, TD-M(PC)^2 and TD-MPC2 fail to generate physical behavior for balancing from the beginning: their legs flick the board away instead of standing on it, as illustrated in Fig. 4d.

5) Navigation through Standing Poles: Despite the high-return rewards of approximately 600 from TD-M(PC)^2 and TD-GRPC, the robot's behavior through these algorithms differs from when learning with SAC and TD-MPC2, where the task requires the robot to move forward while avoiding collision with the poles to gain rewards. Meanwhile, the policies trained on SAC and TD-MPC2 tell the robot to go to the side of the room and follow the room edge, effectively avoiding collision as a safe strategy. However, the robot collides with the poles and falls, unable to accomplish the task. TD-M(PC)^2 and TD-GRPC direct the robot to move forward, go back, and repeat such actions until the episode ends. As a result, the robot does not move much, staying at the same spot, but is still gaining rewards (Fig. 4e).

6) Arm-Balancing Helps Locomotion: Throughout locomotion experiments on `HumanoidBench`, we observe that the robot arms are "*ill-posed*" and not in optimal states. While investigating such behaviors, we find that arm-balancing helps humanoid locomotion despite random arm movements while executing tasks. For instance, we lock both arms at fixed positions, and make it learn a locomotion task (*e.g.*, walking, running). Consequently, the robot could not accomplish the tasks when learning with different algorithms. Fig. 4f reflects their intricacy and barely complete the tasks, where the robots unsteadily fall during their learning episodes. Constraining these arms' poses could further benefit robot actions in real-world scenarios.

C. Demonstration

Besides solving abilities of algorithms learned from quantitative and qualitative results in Tab. II, the demonstration video of our experiments can be seen in the supplementary document for comparisons of H1-2's performance across locomotion tasks between benchmarked algorithms.

V. Conclusions

We presented TD-GRPC – a framework incorporating GRPO, explicit policy constraints, and TD learning for stable policy updates during learning humanoid locomotion tasks. Via this RL-based method, we achieve robust and sample-efficient learning by constraining policy rollouts in latent space without restricting planner flexibility. Our results on `HumanoidBench` with the 26-DoF Unitree H1-2 humanoid demonstrate that TD-GRPC surpasses existing baselines in stability and quantitative and qualitative performances and sets a foundation for scalable, constraint-aware RL in high-dimensional complex humanoid control.

References

[1] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Learning humanoid locomotion with transformers," *CoRR*, 2023.

[2] ——, "Real-world humanoid locomotion with reinforcement learning," *Science Robotics*, vol. 9, no. 89, p. eadi9579, 2024.

[3] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *IEEE ICRA*, 2018.

[4] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," *NeurIPS*, 2018.

[5] A. Argenson and G. Dulac-Arnold, "Model-based offline planning," *arXiv preprint arXiv:2008.05556*, 2020.

[6] B. Kouvaritakis and M. Cannon, "Model predictive control," *Switzerland: Springer International Publishing*, vol. 38, no. 13-56, p. 7, 2016.

[7] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *ICML*, 2018.

[8] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel *et al.*, "Mastering atari, go, chess and shogi by planning with a learned model," *Nature*, vol. 588, no. 7839, pp. 604–609, 2020.

[9] W. Ye, S. Liu, T. Kurutach, P. Abbeel, and Y. Gao, "Mastering atari games with limited data," *NeurIPS*, 2021.

[10] H. Sikchi, W. Zhou, and D. Held, "Learning off-policy with online planning," in *CoRL*, 2022.

[11] N. Hansen, H. Su, and X. Wang, "Td-mpc2: Scalable, robust world models for continuous control," *arXiv preprint arXiv:2310.16828*, 2023.

[12] H. Lin, P. Wang, J. Schneider, and G. Shi, "Improving temporal difference mpc through policy constraint," *arXiv preprint arXiv:2502.03550*, 2025.

[13] R. S. Sutton, "Dyna, an integrated architecture for learning, planning, and reacting," *ACM Sigart Bulletin*, vol. 2, no. 4, pp. 160–163, 1991.

[14] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to trust your model: Model-based policy optimization," *NeurIPS*, 2019.

[15] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine *et al.*, "Model-based reinforcement learning for atari," *arXiv preprint arXiv:1903.00374*, 2019.

[16] T. Pham and A. Cangelosi, "Pay attention to what and where? interpretable feature extractor in vision-based deep reinforcement learning," in *IJCNN*, 2025.

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

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

[19] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[20] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," *arXiv preprint arXiv:1912.01603*, 2019.

[21] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba, "Mastering atari with discrete world models," *arXiv preprint arXiv:2010.02193*, 2020.

[22] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," *arXiv preprint arXiv:2301.04104*, 2023.

[23] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," *NeurIPS*, 2018.

[24] N. Lambert, B. Amos, O. Yadan, and R. Calandra, "Objective mismatch in model-based reinforcement learning," *arXiv preprint arXiv:2002.04523*, 2020.

[25] Y. Xu, N. Hansen, Z. Wang, Y.-C. Chan, H. Su, and Z. Tu, "On the feasibility of cross-task transfer with model-based reinforcement learning," *arXiv preprint arXiv:2210.10763*, 2022.

[26] N. Hansen, X. Wang, and H. Su, "Temporal difference learning for model predictive control," *arXiv preprint arXiv:2203.04955*, 2022.

[27] I. Clavera, V. Fu, and P. Abbeel, "Model-augmented actor-critic: Backpropagating through paths," *arXiv preprint arXiv:2005.08068*, 2020.

[28] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration," in *ICML*, 2019.

[29] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," *arXiv preprint arXiv:2110.06169*, 2021.

[30] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *ICML*, 2017.

[31] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.

[32] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu *et al.*, "Deepseekmath: Pushing the limits of mathematical reasoning in open language models," *arXiv preprint arXiv:2402.03300*, 2024.

[33] C. Sferrazza, D.-M. Huang, X. Lin, Y. Lee, and P. Abbeel, "Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation," *arXiv preprint arXiv:2403.10506*, 2024.

[34] J. Peters, S. Vijayakumar, and S. Schaal, "Reinforcement learning for humanoid robotics," in *IEEE-RAS Humanoids*, 2003.

[35] A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine, "Stabilizing off-policy q-learning via bootstrapping error reduction," *NeurIPS*, vol. 32, 2019.

[36] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," *NeurIPS*, vol. 33, pp. 1179–1191, 2020.

[37] S. Fujimoto and S. S. Gu, "A minimalist approach to offline reinforcement learning," *NeurIPS*, vol. 34, pp. 20 132–20 145, 2021.

[38] X. B. Peng, A. Kumar, G. Zhang, and S. Levine, "Advantage-weighted regression: Simple and scalable off-policy reinforcement learning," *arXiv preprint arXiv:1910.00177*, 2019.

[39] D. Garg, J. Hejna, M. Geist, and S. Ermon, "Extreme q-learning: Maxent rl without entropy," *arXiv preprint arXiv:2301.02328*, 2023.

[40] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *IEEE ICRA*, 2016.

[41] S. P. Singh and R. C. Yee, "An upper bound on the loss from approximate optimal-value functions," *Machine Learning*, vol. 16, pp. 227–233, 1994.

[42] J. Peters, K. Mulling, and Y. Altun, "Relative entropy policy search," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 24, no. 1, 2010, pp. 1607–1612.

[43] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *ICML*, 2015.

[44] A. Nair, A. Gupta, M. Dalal, and S. Levine, "Awac: Accelerating online reinforcement learning with offline datasets," *arXiv preprint arXiv:2006.09359*, 2020.

[45] R. Y. Rubinstein, "Optimization of computer simulation models with rare events," *European Journal of Operational Research*, vol. 99, no. 1, pp. 89–112, 1997.

[46] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey *et al.*, "Maximum entropy inverse reinforcement learning." in *Aaai*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[47] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. d. L. Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq *et al.*, "Deepmind control suite," *arXiv preprint arXiv:1801.00690*, 2018.