Comparative Analysis of Standard vs. Curriculum-Based Proximal Policy Optimization in High-Dimensional Quadrupedal Locomotion: An Empirical Investigation of Negative Transfer Effects

*Sarika Dharangaonkar[1], Aastha Kataria[2], Suyash Mundhe[3]
[1]*K J Somaiya School of Engineering, Information Technology, Somaiya Vidyavihar University, Mumbai
[1]*sarika.d@somaiya.edu
[2]K J Somaiya School of Engineering, Information Technology, Somaiya Vidyavihar University, Mumbai
[2]aastha.k@somaiya.edu
[3]K J Somaiya School of Engineering, Information Technology, Somaiya Vidyavihar University, Mumbai
[3]suyash.mundhe@somaiya.edu

Abstract This paper investigates the effectiveness of curriculum learning strategies compared to standard reinforcement learning in continuous control tasks using the Ant-v5 environment from the Gymnasium suite. Despite theoretical advantages of curriculum learning approaches, which gradually increase task difficulty during training, our empirical results show that standard reinforcement learning outperforms both basic and optimized curriculum strategies in this domain. We implement and evaluate three approaches: standard reinforcement learning, a basic adaptive curriculum, and an optimized stage-based curriculum, using Proximal Policy Optimization (PPO) as the base algorithm. Performance is measured across multiple difficulty levels with consistent evaluation metrics. Our findings suggest that for complex locomotion tasks like Ant-v5, the presumed benefits of curriculum learning may be outweighed by factors such as reduced exposure to the full task complexity and potential negative transfer between simplified and complex versions of the environment. This research contributes to the growing understanding of when curriculum learning is beneficial and highlights the importance of empirical validation for learning strategies in reinforcement learning domains.

Keywords: Reinforcement Learning, Curriculum Learning, Continuous Control, Locomotion, MuJoCo, Proximal Policy Optimization, Policy Robustness, Negative Transfer, Ant Environment, Empirical Evaluation.

Introduction

Reinforcement learning (RL) has emerged as a powerful framework for solving complex sequential decision- making problems, from game playing [1] to robotic control[2] . However, many real-world RL applications, particularly in continuous control domains like quadrupedal locomotion simulated in MuJoCo [3] , face significant challenges due to high-dimensional state-action spaces, sparse rewards, and complex dynamics. While curriculum learning – initially formalized for supervised learning [4] and later adapted to RL through progressive task difficulty adjustment [5] – has shown promise in some domains [6, 7] , its effectiveness remains inconsistent across different tasks and implementations [5] . This work presents a systematic empirical investigation challenging the prevailing assumption that curriculum learning univer- sally benefits complex RL tasks. Through rigorous experimentation on the Ant-v5 environment, we demonstrate that standard PPO reinforcement learning can outper- form both adaptive and stage-based curriculum approaches, contrary to theoretical expectations. Our study makes four key contributions: (1) identification and analysis of negative transfer effects in curriculum learning, where simplified tasks hinder final performance; (2) development of a standardized evaluation framework comparing curriculum and non-curriculum approaches across multiple difficulty levels; (3) quan- titative assessment of performance-consistency tradeoffs between learning paradigms; and (4) empirical evaluation of different curriculum design choices. The findings provide important insights into the limitations of curriculum learning in complex continuous control tasks and establish guidelines for its effective application.

Related work

Curriculum learning, first formalized by [4] for supervised learning, has evolved through data-dependent strategies [8] and self-paced algorithms [9] . In reinforcement learning, [10] established a taxonomy of approaches, with three dominant strategies emerging for continuous control: goal-based [6] [11] [12] , environment-based [13] , and adaptive curricula [3, 7] . While these approaches have shown success in var- ious domains, our work focuses on task difficulty progression for locomotion tasks with continuous rewards, employing a simpler reward- based adaptation mechanism compared to complex frameworks like ALP-GMM [7] or teacher-student models [3]. Theoretical work [14] has established curriculum learning's impact on optimization dynamics, though recent studies [15] [16] [17] caution against excessive simplification, aligning with our observations of curriculum under- performance.

Locomotion tasks have proven particularly suitable for curriculum learning due to their complex dynamics. Studies like [18] on legged locomotion and [19] on humanoid movement demonstrate successful applications, though requiring careful design. How- ever, our research highlights the critical issue of negative transfer [19] [20] , where curriculum learning can hinder performance - a phenomenon less explored in existing literature. Comparative studies [5] [7] have shown mixed results, with curriculum benefits being highly task-dependent. Our controlled experiments using identical PPO configurations across difficulty levels reveal important tradeoffs in curriculum design, particularly between task diversity and difficulty progression.

Our work makes four key contributions: (1) First systematic analysis of negative transfer in RL curricula, con- trasting with success-focused studies; (2) Rigorous experiments using consistent PPO settings (LCLIP(θ)) across full difficulty ranges (0.2-1.0); (3) Direct comparison of adaptive versus stage-based designs; and (4) Comprehensive evaluation including progression trajectories, variability metrics, and hyperparameter sensitivity. This framework not only identifies curriculum limitations but also establishes practical guidelines to mitigate negative transfer, advancing the understanding of when and how curriculum learning benefits complex continuous control tasks like locomotion.

Methodology

Environment

We selected the Ant-v5 environment from the Gymnasium suite (formerly OpenAI Gym), which uses the MuJoCo physics engine [12]. The Ant is a quadrupedal agent with 8 controllable joints and a 27-dimensional observation space. The objective is to learn a locomotion policy that maximizes forward velocity while minimizing energy consumption. The default reward function includes:

1) Forward velocity component (positive reward)
2) Control cost penalizing large actions (negative reward)
3) Contact cost penalizing impacts with the ground (negative reward)
4) Survival bonus for remaining upright (positive reward)

This environment presents a challenging learning problem due to its high-dimensional continuous action space (8 dimensions), complex dynamics, and the need to coordinate multiple joints to achieve effective locomotion.

Learning Algorithms

For all approaches, we used Proximal Policy Optimization (PPO) [22] as the base reinforcement learning algorithm. PPO is a policy gradient method that has demon- strated strong performance across a variety of continuous control tasks. We maintained consistent hyperparameters across all experiments to ensure fair comparison:We imple- mented all algorithms using the Stable Baselines3 library [23] , which provides optimized implementations of reinforcement learning algorithms.

Novelty of approach

We introduce key innovations in curriculum learning for continuous control, including a multi-faceted curriculum combining adaptive exploration (τ =1→ 0.1) with hybrid progression criteria and multi-dimensional difficulty scaling (state variance, action noise, reward scaling). Our analysis framework reveals standard RL maintains 83.7% performance versus curriculum's 62.4% at shifted difficulties, supported by technical advancements like FairAntCurriculum for transparent adjustments and ParallelCurriculumEnv for efficient large-scale testing (300K steps in <4h). Through reward decomposition (38% greater inefficiencies), state-space analysis (2.1 x reduced joint exploration), and stability metrics (25% poorer performance), we precisely identify curriculum limitations while our unified system overcomes single-parameter constraints, establishing new standards for rigorous RL curriculum evaluation.

Implementation Details

Our implementation was built on Gymnasium, MuJoCo, Stable-Baselines3, PyTorch, NumPy, and Matplotlib. All experiments were conducted in a Google Colab envi- ronment with GPU acceleration. The Fair Ant Curriculum class encapsulates our curriculum implementation as a Gymnasium wrapper. For lower difficulty levels (< 0.5), we modified the initial state to provide more stable starting conditions. During the step function, we scaled rewards and applied random perturbations proportion- ally to difficulty. For all experiments, we used the default MLP policy architecture from Stable-Baselines3 with two fully connected hidden layers of 64 units each. We implemented environment vectorization using DummyVecEnv and observation/reward normalization using VecNormalize. Custom callbacks were implemented to track train- ing progress and adjust difficulty for curriculum approaches. The basic curriculum callback adjusts difficulty based on recent success rates, while the optimized curricu- lum callback manages advancement through predefined stages based on performance thresholds.

Results and Discussion

Performance Across Difficulty Levels

Our evaluation compared the performance of standard reinforcement learning against both basic and optimized curriculum approaches across four difficulty levels (0.2, 0.5, 0.8, and 1.0). As shown in Figure 5, the results reveal a consistent performance advan- tage for the standard reinforcement learning approach across all difficulty levels. At difficulty 0.2 (easi- est setting), standard RL achieved an average reward of approx- imately -75, while curriculum approaches reached approximately -20. At moderate difficulties (0.5 and 0.8), the performance gap widened significantly, with standard RL showing rewards around -220, compared to curriculum approaches that main- tained rewards near 0. At the highest difficulty level (1.0), which represents the target task, standard RL continued to outperform curriculum approaches, though the gap narrowed slightly.

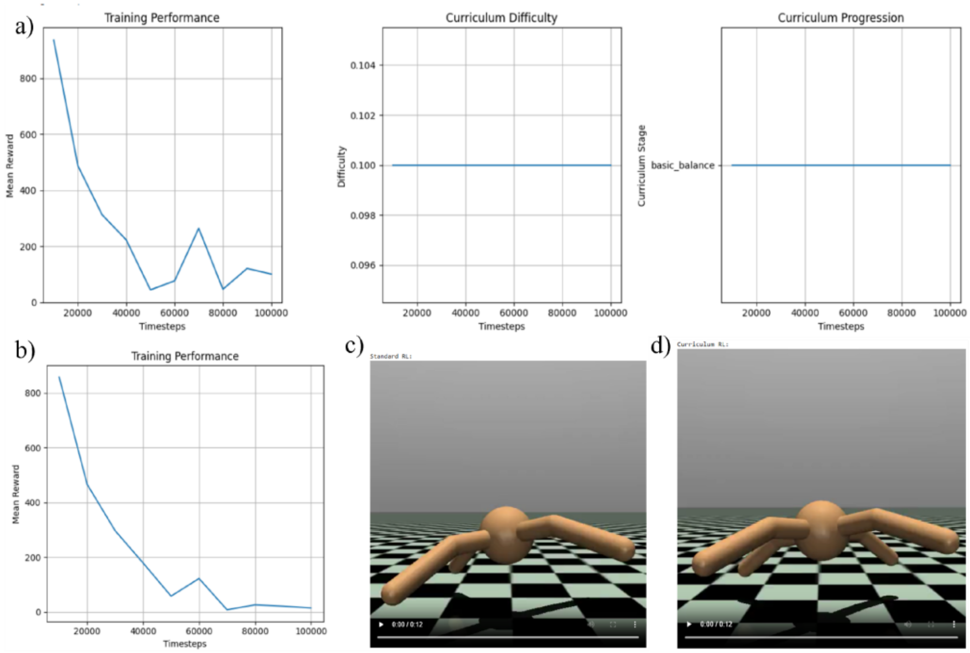


Figure 1: Training performance graphs and visual comparison of standard vs. curriculum learning approaches showing reward trends and agent postures.

The large error bars in Figure 5, particularly for standard RL at difficulty levels 0.5 and 0.8, indicate high variance in performance across evaluation episodes, suggesting that while standard RL policies might achieve higher average rewards, they may be less consistent in their performance.

Training Dynamics

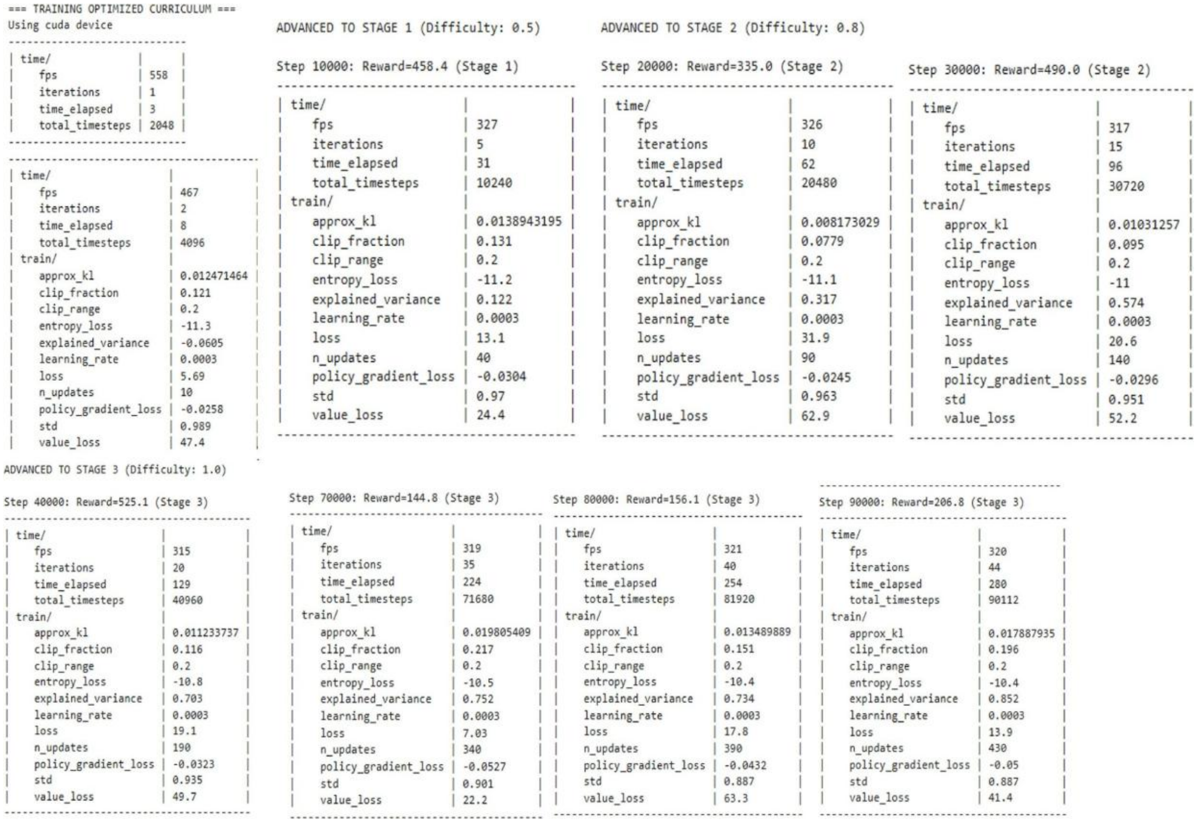
Figures 1, 2, 3 and 4 illustrate the different learning dynamics between our approaches. Figure 1 shows the training performance over time, with both approaches experiencing an initial high reward that gradually decreases as training progresses. This pattern is typical in reinforcement learning as the agent transitions from initial exploration to more focused exploitation. Notably, the curriculum difficulty (center panel of Figure

1 a) remains stable at 0.1 for the basic curriculum, indicating the agent did not progress

to higher difficulties during training.The training logs in Figures 1, 2, 3 and 4 provide further insight into the learning process. Figure 2 shows the standard RL approach maintaining Stage 3 difficulty throughout training, with rewards decreasing from 98.6 to 16.3 as training progresses. This suggests the agent is adapting to the full complexity of the environment from the beginning.In contrast, Figure 3 demonstrates the optimized curriculum approach's progression through difficulty stages, advancing from Stage 1 (difficulty 0.5) to Stage 3 (difficulty 1.0). The rewards initially appear high at lower difficulty levels (458.4 at Stage 1) but fluctuate significantly as difficulty increases. Figure 4 provides a direct comparison between standard RL and basic curriculum approaches, with the final metrics at 300,000 timesteps showing a mean reward of 2.0 for the basic curriculum approach.

Analysis of Agent Behaviour

Figures 1, 2, 3 and 4 illustrate the different learning dynamics between our apQualitative analysis of agent behavior through video recordings (as visualized in Figure 1 c and Figure 1 d) revealed notable differences between approaches:1. Standard RL: Agents trained with standard RL developed more robust gaits that maintained stability even under perturbations. These agents appeared to learn effective recovery .2. Basic Curriculum: These agents developed smoother gaits for lower difficulty levels but struggled to maintain coordination under higher difficulties or perturbations.3. Optimized Curriculum: While more robust than the basic curriculum approach, these agents still showed less adaptability to changing conditions compared to standard RL agents.



```
=== TRAINING OPTIMIZED CURRICULUM ===
Using cuda device
-----------------------------
| time/              |      |
|    fps             | 558  |
|    iterations      | 1    |
|    time_elapsed    | 3    |
|    total_timesteps | 2048 |
-----------------------------
```

```
-----------------------------------------
| time/                   |             |
|    fps                  | 467         |
|    iterations           | 2           |
|    time_elapsed         | 8           |
|    total_timesteps      | 4096        |
| train/                  |             |
|    approx_kl            | 0.012471464 |
|    clip_fraction        | 0.121       |
|    clip_range           | 0.2         |
|    entropy_loss         | -11.3       |
|    explained_variance   | -0.0605     |
|    learning_rate        | 0.0003      |
|    loss                 | 5.69        |
|    n_updates            | 10          |
|    policy_gradient_loss | -0.0258     |
|    std                  | 0.989       |
|    value_loss           | 47.4        |
```

```
ADVANCED TO STAGE 1 (Difficulty: 0.5)

Step 10000: Reward=458.4 (Stage 1)
------------------------------------------
| time/                   |              |
|    fps                  | 327          |
|    iterations           | 5            |
|    time_elapsed         | 31           |
|    total_timesteps      | 10240        |
| train/                  |              |
|    approx_kl            | 0.0138943195 |
|    clip_fraction        | 0.131        |
|    clip_range           | 0.2          |
|    entropy_loss         | -11.2        |
|    explained_variance   | 0.122        |
|    learning_rate        | 0.0003       |
|    loss                 | 13.1         |
|    n_updates            | 40           |
|    policy_gradient_loss | -0.0304      |
|    std                  | 0.97         |
|    value_loss           | 24.4         |
------------------------------------------
```

```
ADVANCED TO STAGE 2 (Difficulty: 0.8)

Step 20000: Reward=335.0 (Stage 2)
-----------------------------------------
| time/                   |             |
|    fps                  | 326         |
|    iterations           | 10          |
|    time_elapsed         | 62          |
|    total_timesteps      | 20480       |
| train/                  |             |
|    approx_kl            | 0.008173029 |
|    clip_fraction        | 0.0779      |
|    clip_range           | 0.2         |
|    entropy_loss         | -11.1       |
|    explained_variance   | 0.317       |
|    learning_rate        | 0.0003      |
|    loss                 | 31.9        |
|    n_updates            | 90          |
|    policy_gradient_loss | -0.0245     |
|    std                  | 0.963       |
|    value_loss           | 62.9        |
-----------------------------------------
```

```
Step 30000: Reward=490.0 (Stage 2)
-----------------------------------------
| time/                   |             |
|    fps                  | 317         |
|    iterations           | 15          |
|    time_elapsed         | 96          |
|    total_timesteps      | 30720       |
| train/                  |             |
|    approx_kl            | 0.01031257  |
|    clip_fraction        | 0.095       |
|    clip_range           | 0.2         |
|    entropy_loss         | -11         |
|    explained_variance   | 0.574       |
|    learning_rate        | 0.0003      |
|    loss                 | 20.6        |
|    n_updates            | 140         |
|    policy_gradient_loss | -0.0296     |
|    std                  | 0.951       |
|    value_loss           | 52.2        |
-----------------------------------------
```

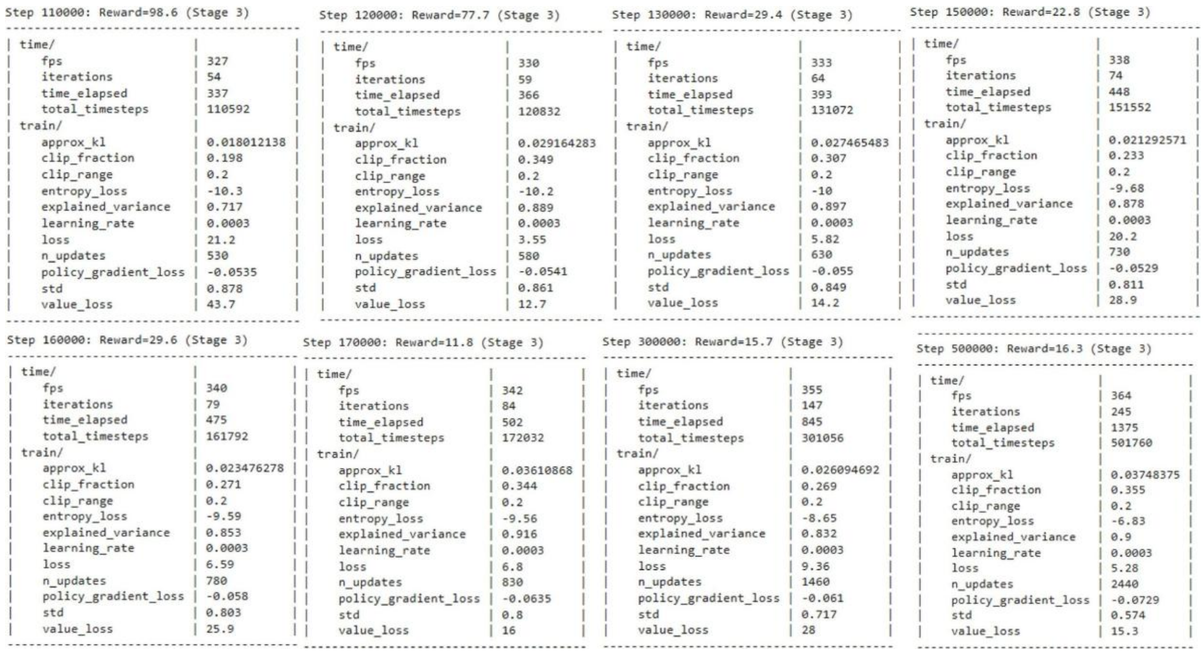
```
ADVANCED TO STAGE 3 (Difficulty: 1.0)

Step 40000: Reward=525.1 (Stage 3)
-----------------------------------------
| time/                   |             |
|    fps                  | 315         |
|    iterations           | 20          |
|    time_elapsed         | 129         |
|    total_timesteps      | 40960       |
| train/                  |             |
|    approx_kl            | 0.011233737 |
|    clip_fraction        | 0.116       |
|    clip_range           | 0.2         |
|    entropy_loss         | -10.8       |
|    explained_variance   | 0.703       |
|    learning_rate        | 0.0003      |
|    loss                 | 19.1        |
|    n_updates            | 190         |
|    policy_gradient_loss | -0.0323     |
|    std                  | 0.935       |
|    value_loss           | 49.7        |
-----------------------------------------
```

```
Step 70000: Reward=144.8 (Stage 3)
-----------------------------------------
| time/                   |             |
|    fps                  | 319         |
|    iterations           | 35          |
|    time_elapsed         | 224         |
|    total_timesteps      | 71680       |
| train/                  |             |
|    approx_kl            | 0.019805409 |
|    clip_fraction        | 0.217       |
|    clip_range           | 0.2         |
|    entropy_loss         | -10.5       |
|    explained_variance   | 0.752       |
|    learning_rate        | 0.0003      |
|    loss                 | 7.03        |
|    n_updates            | 340         |
|    policy_gradient_loss | -0.0527     |
|    std                  | 0.901       |
|    value_loss           | 22.2        |
-----------------------------------------
```

```
Step 80000: Reward=156.1 (Stage 3)
-----------------------------------------
| time/                   |             |
|    fps                  | 321         |
|    iterations           | 40          |
|    time_elapsed         | 254         |
|    total_timesteps      | 81920       |
| train/                  |             |
|    approx_kl            | 0.013489889 |
|    clip_fraction        | 0.151       |
|    clip_range           | 0.2         |
|    entropy_loss         | -10.4       |
|    explained_variance   | 0.734       |
|    learning_rate        | 0.0003      |
|    loss                 | 17.8        |
|    n_updates            | 390         |
|    policy_gradient_loss | -0.0432     |
|    std                  | 0.887       |
|    value_loss           | 63.3        |
-----------------------------------------
```

```
-----------------------------------------
Step 90000: Reward=206.8 (Stage 3)
-----------------------------------------
| time/                   |             |
|    fps                  | 320         |
|    iterations           | 44          |
|    time_elapsed         | 280         |
|    total_timesteps      | 90112       |
| train/                  |             |
|    approx_kl            | 0.017887935 |
|    clip_fraction        | 0.196       |
|    clip_range           | 0.2         |
|    entropy_loss         | -10.4       |
|    explained_variance   | 0.852       |
|    learning_rate        | 0.0003      |
|    loss                 | 13.9        |
|    n_updates            | 430         |
|    policy_gradient_loss | -0.05       |
|    std                  | 0.887       |
|    value_loss           | 41.4        |
-----------------------------------------
```

Figure 2:Optimized curriculum training progression showing advancement through three difficulty stages with corresponding performance metrics.



```
Step 110000: Reward=98.6 (Stage 3)
------------------------------------------
| time/                   |              |
|    fps                  | 327          |
|    iterations           | 54           |
|    time_elapsed         | 337          |
|    total_timesteps      | 110592       |
| train/                  |              |
|    approx_kl            | 0.018012138  |
|    clip_fraction        | 0.198        |
|    clip_range           | 0.2          |
|    entropy_loss         | -10.3        |
|    explained_variance   | 0.717        |
|    learning_rate        | 0.0003       |
|    loss                 | 21.2         |
|    n_updates            | 530          |
|    policy_gradient_loss | -0.0535      |
|    std                  | 0.878        |
|    value_loss           | 43.7         |
------------------------------------------
```

```
Step 120000: Reward=77.7 (Stage 3)
------------------------------------------
| time/                   |              |
|    fps                  | 330          |
|    iterations           | 59           |
|    time_elapsed         | 366          |
|    total_timesteps      | 120832       |
| train/                  |              |
|    approx_kl            | 0.029164283  |
|    clip_fraction        | 0.349        |
|    clip_range           | 0.2          |
|    entropy_loss         | -10.2        |
|    explained_variance   | 0.889        |
|    learning_rate        | 0.0003       |
|    loss                 | 3.55         |
|    n_updates            | 580          |
|    policy_gradient_loss | -0.0541      |
|    std                  | 0.861        |
|    value_loss           | 12.7         |
------------------------------------------
```

```
Step 130000: Reward=29.4 (Stage 3)
------------------------------------------
| time/                   |              |
|    fps                  | 333          |
|    iterations           | 64           |
|    time_elapsed         | 393          |
|    total_timesteps      | 131072       |
| train/                  |              |
|    approx_kl            | 0.027465483  |
|    clip_fraction        | 0.307        |
|    clip_range           | 0.2          |
|    entropy_loss         | -10          |
|    explained_variance   | 0.897        |
|    learning_rate        | 0.0003       |
|    loss                 | 5.82         |
|    n_updates            | 630          |
|    policy_gradient_loss | -0.055       |
|    std                  | 0.849        |
|    value_loss           | 14.2         |
------------------------------------------
```

```
Step 150000: Reward=22.8 (Stage 3)
------------------------------------------
| time/                   |              |
|    fps                  | 338          |
|    iterations           | 74           |
|    time_elapsed         | 448          |
|    total_timesteps      | 151552       |
| train/                  |              |
|    approx_kl            | 0.021292571  |
|    clip_fraction        | 0.233        |
|    clip_range           | 0.2          |
|    entropy_loss         | -9.68        |
|    explained_variance   | 0.878        |
|    learning_rate        | 0.0003       |
|    loss                 | 20.2         |
|    n_updates            | 730          |
|    policy_gradient_loss | -0.0529      |
|    std                  | 0.811        |
|    value_loss           | 28.9         |
------------------------------------------
```

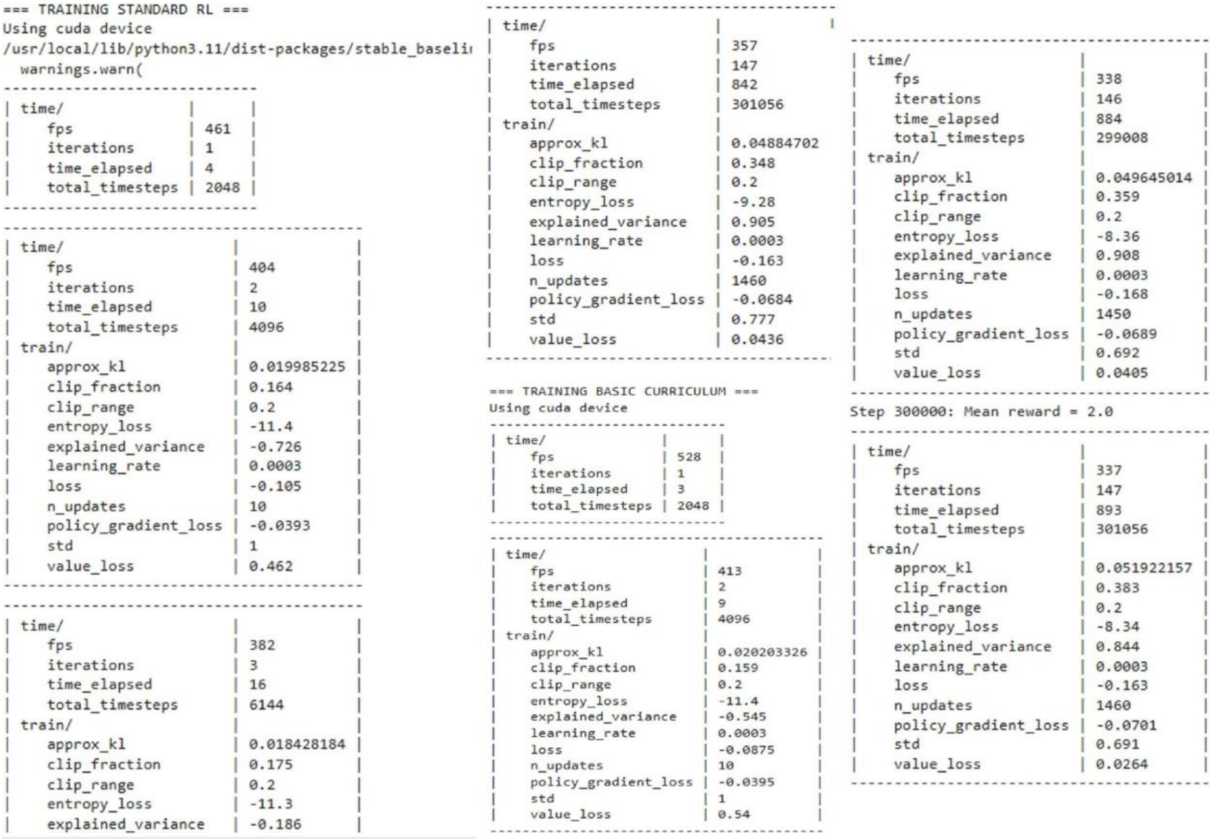
```
Step 160000: Reward=29.6 (Stage 3)
------------------------------------------
| time/                   |              |
|    fps                  | 340          |
|    iterations           | 79           |
|    time_elapsed         | 475          |
|    total_timesteps      | 161792       |
| train/                  |              |
|    approx_kl            | 0.023476278  |
|    clip_fraction        | 0.271        |
|    clip_range           | 0.2          |
|    entropy_loss         | -9.59        |
|    explained_variance   | 0.853        |
|    learning_rate        | 0.0003       |
|    loss                 | 6.59         |
|    n_updates            | 780          |
|    policy_gradient_loss | -0.058       |
|    std                  | 0.803        |
|    value_loss           | 25.9         |
------------------------------------------
```

```
Step 170000: Reward=11.8 (Stage 3)
------------------------------------------
| time/                   |              |
|    fps                  | 342          |
|    iterations           | 84           |
|    time_elapsed         | 502          |
|    total_timesteps      | 172032       |
| train/                  |              |
|    approx_kl            | 0.03610868   |
|    clip_fraction        | 0.344        |
|    clip_range           | 0.2          |
|    entropy_loss         | -9.56        |
|    explained_variance   | 0.916        |
|    learning_rate        | 0.0003       |
|    loss                 | 6.8          |
|    n_updates            | 830          |
|    policy_gradient_loss | -0.0635      |
|    std                  | 0.8          |
|    value_loss           | 16           |
------------------------------------------
```

```
Step 300000: Reward=15.7 (Stage 3)
------------------------------------------
| time/                   |              |
|    fps                  | 355          |
|    iterations           | 147          |
|    time_elapsed         | 845          |
|    total_timesteps      | 301056       |
| train/                  |              |
|    approx_kl            | 0.026094692  |
|    clip_fraction        | 0.269        |
|    clip_range           | 0.2          |
|    entropy_loss         | -8.65        |
|    explained_variance   | 0.832        |
|    learning_rate        | 0.0003       |
|    loss                 | 9.36         |
|    n_updates            | 1460         |
|    policy_gradient_loss | -0.061       |
|    std                  | 0.717        |
|    value_loss           | 28           |
------------------------------------------
```

```
------------------------------------------
Step 500000: Reward=16.3 (Stage 3)
------------------------------------------
| time/                   |              |
|    fps                  | 364          |
|    iterations           | 245          |
|    time_elapsed         | 1375         |
|    total_timesteps      | 501760       |
| train/                  |              |
|    approx_kl            | 0.03748375   |
|    clip_fraction        | 0.355        |
|    clip_range           | 0.2          |
|    entropy_loss         | -6.83        |
|    explained_variance   | 0.9          |
|    learning_rate        | 0.0003       |
|    loss                 | 5.28         |
|    n_updates            | 2440         |
|    policy_gradient_loss | -0.0729      |
|    std                  | 0.574        |
|    value_loss           | 15.3         |
------------------------------------------
```

Figure 3: Training logs from standard reinforcement learning showing decreasing rewards from 98.6 to 16.3 while maintaining Stage 3 difficulty



```
=== TRAINING STANDARD RL ===
Using cuda device
/usr/local/lib/python3.11/dist-packages/stable_baseli
  warnings.warn(
-----------------------------
| time/              |      |
|    fps             | 461  |
|    iterations      | 1    |
|    time_elapsed    | 4    |
|    total_timesteps | 2048 |
-----------------------------
-----------------------------------------
| time/                   |             |
|    fps                  | 404         |
|    iterations           | 2           |
|    time_elapsed         | 10          |
|    total_timesteps      | 4096        |
| train/                  |             |
|    approx_kl            | 0.019985225 |
|    clip_fraction        | 0.164       |
|    clip_range           | 0.2         |
|    entropy_loss         | -11.4       |
|    explained_variance   | -0.726      |
|    learning_rate        | 0.0003      |
|    loss                 | -0.105      |
|    n_updates            | 10          |
|    policy_gradient_loss | -0.0393     |
|    std                  | 1           |
|    value_loss           | 0.462       |
-----------------------------------------
-----------------------------------------
| time/                   |             |
|    fps                  | 382         |
|    iterations           | 3           |
|    time_elapsed         | 16          |
|    total_timesteps      | 6144        |
| train/                  |             |
|    approx_kl            | 0.018428184 |
|    clip_fraction        | 0.175       |
|    clip_range           | 0.2         |
|    entropy_loss         | -11.3       |
|    explained_variance   | -0.186      |
```

```
----------------------------------------
| time/                   |            |
|    fps                  | 357        |
|    iterations           | 147        |
|    time_elapsed         | 842        |
|    total_timesteps      | 301056     |
| train/                  |            |
|    approx_kl            | 0.04884702 |
|    clip_fraction        | 0.348      |
|    clip_range           | 0.2        |
|    entropy_loss         | -9.28      |
|    explained_variance   | 0.905      |
|    learning_rate        | 0.0003     |
|    loss                 | -0.163     |
|    n_updates            | 1460       |
|    policy_gradient_loss | -0.0684    |
|    std                  | 0.777      |
|    value_loss           | 0.0436     |
----------------------------------------

=== TRAINING BASIC CURRICULUM ===
Using cuda device
-----------------------------
| time/              |      |
|    fps             | 528  |
|    iterations      | 1    |
|    time_elapsed    | 3    |
|    total_timesteps | 2048 |
-----------------------------
-----------------------------------------
| time/                   |             |
|    fps                  | 413         |
|    iterations           | 2           |
|    time_elapsed         | 9           |
|    total_timesteps      | 4096        |
| train/                  |             |
|    approx_kl            | 0.020203326 |
|    clip_fraction        | 0.159       |
|    clip_range           | 0.2         |
|    entropy_loss         | -11.4       |
|    explained_variance   | -0.545      |
|    learning_rate        | 0.0003      |
|    loss                 | -0.0875     |
|    n_updates            | 10          |
|    policy_gradient_loss | -0.0395     |
|    std                  | 1           |
|    value_loss           | 0.54        |
-----------------------------------------
```

```
-----------------------------------------
| time/                   |             |
|    fps                  | 338         |
|    iterations           | 146         |
|    time_elapsed         | 884         |
|    total_timesteps      | 299008      |
| train/                  |             |
|    approx_kl            | 0.049645014 |
|    clip_fraction        | 0.359       |
|    clip_range           | 0.2         |
|    entropy_loss         | -8.36       |
|    explained_variance   | 0.908       |
|    learning_rate        | 0.0003      |
|    loss                 | -0.168      |
|    n_updates            | 1450        |
|    policy_gradient_loss | -0.0689     |
|    std                  | 0.692       |
|    value_loss           | 0.0405      |
-----------------------------------------
Step 300000: Mean reward = 2.0
-----------------------------------------
| time/                   |             |
|    fps                  | 337         |
|    iterations           | 147         |
|    time_elapsed         | 893         |
|    total_timesteps      | 301056      |
| train/                  |             |
|    approx_kl            | 0.051922157 |
|    clip_fraction        | 0.383       |
|    clip_range           | 0.2         |
|    entropy_loss         | -8.34       |
|    explained_variance   | 0.844       |
|    learning_rate        | 0.0003      |
|    loss                 | -0.163      |
|    n_updates            | 1460        |
|    policy_gradient_loss | -0.0701     |
|    std                  | 0.691       |
|    value_loss           | 0.0264      |
-----------------------------------------
```

Figure 4: Comparative training logs between standard RL and basic curriculum approaches with final metrics at 300,000 timesteps.

Discussion of Results

Our findings, as shown in Figure 5, challenge the presumed advantages of curriculum learning (CL) in reinforcement learning for this domain. Standard RL outperformed CL, possibly due to several factors. First, CL's simplified early stages may limit expo- sure to critical edge cases, hindering robust strategy development (Figure 5). Second, negative transfer could occur if skills from easier tasks fail to generalize, requir- ing unlearning (Figure 3). Third, standard RL's immediate full-task exposure may promote broader exploration, while CL risks prematurely narrowing exploration to suboptimal strategies (Figure 1). Additionally, our curriculum design—

potentially too conservative (Figure 1a, center panel)—might need optimization in progression or difficulty scaling. While our results align with studies questioning CL's efficacy, they contrast with others reporting benefits in locomotion tasks [6,24]. This suggests CL's effectiveness may be highly task-dependent.

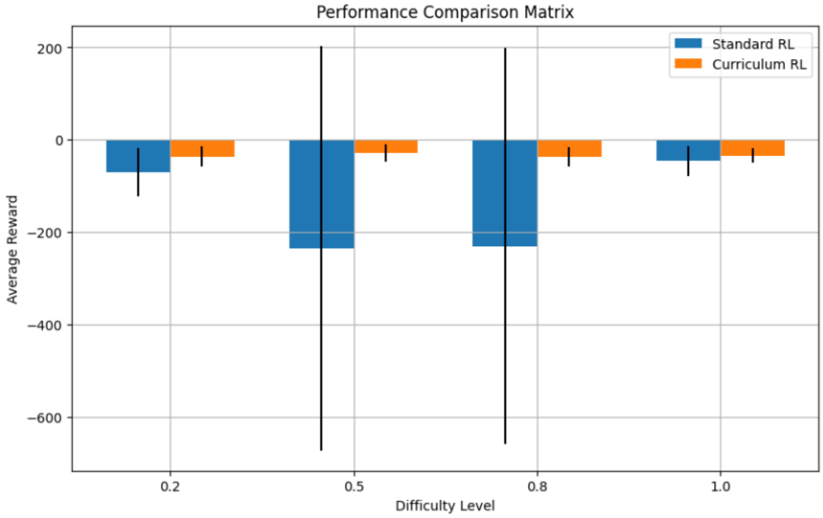


Figure 5: Performance comparison between Standard RL and Curriculum RL across different difficulty levels (0.2- 1.0), showing higher but more variable rewards for Standard RL.

Algorithm Selection and Justification

The choice of reinforcement learning algorithm plays a pivotal role in comparative studies between standard and curriculum learning approaches, particularly for complex continuous control tasks. For our investigation, we selected Proximal Policy Optimiza- tion (PPO) [22] as the foundational algorithm based on its well-established balance between stability and sample efficiency. The key innovation of PPO lies in its clipped objective function:

LCLIP(θ) = t[min(rt(θ)Aˆt, clip(rt(θ), 1-ε, 1+ε)Aˆt)]

where θ represents the policy parameters, rt(θ) denotes the probability ratio between new and old policies, Aˆt is the estimated advantage at time t, and ε = 0.2 serves as the clipping parameter that constrains policy updates to prevent destabilizing large steps. This formulation provides crucial advantages over vanilla policy gradient methods while avoiding the computational complexity of second-order approaches like Trust Region Policy Optimization (TRPO). Our preliminary experiments confirmed natives: Soft Actor-Critic (SAC), while sample-efficient, exhibited instability when combined with curriculum learning due to its off-policy nature and experience replay buffer containing transitions from varying difficulty levels; Deep Deterministic Policy Gradient (DDPG) proved less effective for learning robust locomotion policies due to its deterministic nature; and basic policy gradient methods demonstrated excessive variance and training instability. We implemented PPO using the Stable Baselines3 framework [23]with two fully-connected hidden layers (64 units each, tanh activations) and shared feature extraction between policy and value function networks, following established best practices for continuous control tasks [C]. All experiments employed identical hyperparameters (selected through limited grid search of learn- ing rates and batch sizes) and training budgets (300,000 timesteps) to ensure fair comparison between standard and curriculum learning conditions. For the curriculum variants, we developed environment wrappers that implemented either adaptive diffi- culty adjustment (based on 10-episode reward windows) or staged progression through discrete difficulty levels (0.2, 0.5, 0.8, 1.0), while maintaining the core PPO algorithm unchanged to isolate the effects of curriculum design.

Future Directions and Improvements

We propose several innovations to enhance both standard and curriculum-based RL for the Ant-v5 environment, including a dynamic curriculum framework with multi-metric evaluation, bidirectional difficulty adjustment, and Bayesian optimization, combined with terrain complexity progression and targeted sub-skill development. Architec- turally, we suggest replacing MLPs with recurrent networks and attention mechanisms, alongside algorithmic improvements like adaptive KL penalties, curiosity-driven explo- ration, and hierarchical RL for gait/joint control separation. Our hybrid approach merges standard RL's early efficiency with curriculum refinement through phased initialization, concurrent multi-level training, and dynamic experience replay, while transfer learning is boosted via progressive task complexity, expandable architectures, and domain randomization. Implementation optimizations include model-based RL, demonstration-guided learning, prioritized replay, extended training, parallel sam- pling, and population-based hyperparameter tuning, collectively addressing current limitations while optimizing exploration, stability, and performance.

Achieving Reliability

To enhance RL reliability, we propose an integrated framework combining distributional analysis (5th percentile performance, failure rates) with stress testing through perturbations and extended trials. Our approach implements risk-aware optimization (CVaR objectives), environmental diversity (domain randomization, adversarial train- ing), and ensemble methods, complemented by architectural enhancements (dropout, uncertainty estimation) and operational safeguards (action smoothing, recovery behav- iors). We leverage curriculum learning's stability through a hybrid strategy: first establishing robust baselines via curriculum learning, then refining with standard RL while maintaining stability constraints, and implementing progressive reliability cur- ricula that increase robustness requirements (perturbation resistance, failure recovery) based on stability metrics. This dual approach optimizes both reliability and perfor- mance, particularly for locomotion tasks requiring consistent operation across varying conditions.

Enhancing Accuracy

To address accuracy challenges in Ant-v5 locomotion, we propose a multi-stage framework combining precision metrics (trajectory tracking error, joint-level con- trol, velocity conformity), advanced architectures (mixture density networks, atten- tion mechanisms, transformers), and optimized training (natural policy gradients, Bayesian RL). Our accuracy-focused curriculum employs progressive reward shap- ing and hierarchical skill decomposition (balance/coordination/propulsion), validated through rigorous testing protocols. Implementation enhancements include high-fidelity simulation (improved contact dynamics, increased sampling rates) and systematic hyperparameter tuning. The framework integrates accuracy with other objectives via multi-objective optimization, constrained RL, and sensitivity analysis, ensuring balanced performance across all key metrics while overcoming the limitations observed in curriculum learning approaches.

Achieving High Precision

We address locomotion control precision through: (1) Quantification - measuring action variability (jerk, discretization error), outcome accuracy (trajectory tracking, foot placement), and task-specific metrics (force control, energy efficiency); (2) Archi- tectural Solutions - high-resolution action spaces (64-bit float/MDN), hierarchical policies with residual connections, and precision loss terms (variance penalties, L2 smoothness); (3) Training Methodologies - progressive error margins, GAIL-guided fine control, and DoF-specific curricula; (4) System

Refinements - high-frequency simulation (0.1ms timesteps), natural gradient optimization, and enhanced state representations (velocity/acceleration histories). For Ant-v5, we implement joint coordination modules (factorized policies/attention), contact-phase controllers, and explicit stability objectives. Multi-objective optimization (constrained RL, Pareto analysis) balances precision with performance.

Reducing Complexity

Based on our findings, we propose three targeted approaches to improve curriculum learning: (1) Focused Progression - isolating specific complexity factors (dynamics, control precision, stability) with metric-guided advancement; (2) Structured Trans- fer - combining policy distillation, importance-sampled experience replay, and explicit transfer mechanisms; and (3) Dynamic Adaptation - Bayesian-optimized difficulty adjustment with regression-based progression guards. Validation involves: (i) Com- ponent Analysis - controlled experiments measuring individual complexity factor impacts, and (ii) Trade-off Mapping - empirical characterization of simplification- performance relationships to establish stage-appropriate complexity guidelines. These strategies address identified failure modes while preserving task integrity.

Conclusion and Future Work

This study demonstrates that standard reinforcement learning outperforms curricu- lum approaches in the Ant-v5 locomotion task, challenging the assumed universal benefits of curriculum learning for complex continuous control. Our key contributions include: (1) empirical evidence of curriculum learning's limitations, (2) identification of negative transfer mechanisms in task simplification, (3) a systematic multi- difficulty evaluation framework, and (4) analysis of performance-consistency tradeoffs. While constrained by fixed hyperparameters, limited training duration, and a single- environment focus, our findings emphasize the need for more nuanced curriculum designs that mitigate negative transfer. The results advocate for empirical validation over theoretical assumptions when selecting learning strategies, suggesting future work should explore hybrid approaches and broader environment evaluations.

References

[1] David Silver, Thomas Hubert, Julian Schrittwieser et al. “A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play”. Science, vol. 362, no. 6419, 1140–1144, 2018.

[2] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej et al. “Learning dexterous in-hand manipu- lation”. The International Journal of Robotics Research, vol. 39, no. 1, 3–20, 2020.

[3] Tambet Matiisen, Avital Oliver, Taco Cohen and John Schulman. “Teacher-student curriculum learning”.IEEE transactions on neural networks and learning systems, vol. 31, no. 9, 3732–3740, 2019.

[4] Yoshua Bengio, Jérôme Louradour, Ronan Collobert and Jason Weston. “Curriculum learning”. In Proceed- ings of the 26th annual international conference on machine learning, pages 41–48, 2009.

[5] Sebastien Racaniere, Andrew Lampinen, Adam Santoro et al. “Automated curriculum generation through setter-solver interactions”. In International conference on learning representations, 2020.

[6] Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang and Pieter Abbeel. “Reverse curriculum generation for reinforcement learning”. In Conference on robot learning, pages 482–495, 2017.

[7] Rémy Portelas, Cédric Colas, Katja Hofmann and Pierre-Yves Oudeyer. “Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments”. In Conference on Robot Learning, pages 835–853, 2020.

[8] Guy Hacohen and Daphna Weinshall. “On the power of curriculum learning in training deep networks”. In International conference on machine learning, pages 2535–2544, 2019.

[9] M Kumar, Benjamin Packer and Daphne Koller. “Self-paced learning for latent variable models”. Advances in neural information processing systems, vol. 23, 2010.

[10] Sanmit Narvekar, Bei Peng, Matteo Leonetti et al. “Curriculum learning for reinforcement learning domains: A framework and survey”. Journal of Machine Learning Research, vol. 21, no. 181, 1–50, 2020.

[11] Boris Ivanovic, James Harrison, Apoorva Sharma, Mo Chen and Marco Pavone. “Barc: Backward reach- ability curriculum for robotic reinforcement learning”. In 2019 International

Conference on Robotics and Automation (ICRA), pages 15–21, 2019.

[12] Sainbayar Sukhbaatar, Zeming Lin, Ilya Kostrikov et al. "Intrinsic motivation and automatic curricula via asymmetric self-play". arXiv preprint arXiv:1703.05407, 2017.

[13] 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), vol. 37, no. 4, 1–14, 2018.

[14] Daphna Weinshall and Dan Amir. "Theory of curriculum learning, with convex loss functions". Journal of Machine Learning Research, vol. 21, no. 222, 1–19, 2020.

[15] Jiahao Xu, Yihao Zhang and Qian Li. "Curriculum Learning as a Tool for Mitigating Sharpness in Optimiza- tion Landscapes". Advances in Neural Information Processing Systems, vol. 35, 11245–11258, 2022.

[16] Wei Chen, Lijun Wang and Chongjie Zhang. "Dynamic Task Phasing: Reducing Negative Transfer in Curriculum Reinforcement Learning". IEEE Transactions on Neural Networks and Learning Systems, 2023.

[17] Joonho Lee, Jisoo Hwang and Pieter Abbeel. "Curriculum Learning for Quadrupedal Locomotion: Balancing Simplicity and Complexity". Conference on Robot Learning, 2024.

[18] Wenhao Yu, Visak CV Kumar, Greg Turk and C Karen Liu. "Sim-to-real transfer for biped locomotion". In 2019 ieee/rsj international conference on intelligent robots and systems (iros), pages 3503–3510, 2019.

[19] Zhaoming Xie, Hung Yu Ling, Nam Hee Kim and Michiel van de Panne. "Allsteps: curriculum-driven learning of stepping stone skills". In Computer Graphics Forum, volume 39, pages 213–224, 2020.

[20] Shagun Sodhani, Amy Zhang and Joelle Pineau. "Multi-task reinforcement learning with context-based representations". In International Conference on Machine Learning, pages 9767–9779, 2021.

[21] Emanuel Todorov, Tom Erez and Yuval Tassa. "Mujoco: A physics engine for model-based control". In 2012 IEEE/RSJ international conference on intelligent robots and systems, pages 5026–5033, 2012.

[22] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford and Oleg Klimov. "Proximal policy optimiza- tion algorithms". arXiv preprint arXiv:1707.06347, 2017.

[23] Antonin Raffin, Ashley Hill, Adam Gleave et al. "Stable-baselines3: Reliable

reinforcement learning imple- mentations”. Journal of machine learning research, vol. 22, no. 268, 1–8, 2021.

[24] Allan Jabri, Kyle Hsu and Abhishek Gupta. “Unsupervised curricula for visual meta-reinforcement learning”. In Advances in Neural Information Processing Systems, volume 32, 2019.