

# Reward Training Wheels: Adaptive Auxiliary Rewards for Robotics Reinforcement Learning

Linji Wang\*, Tong Xu\*, Yuanjie Lu, and Xuesu Xiao

**Abstract**—Robotics Reinforcement Learning (RL) often relies on carefully engineered auxiliary rewards to supplement sparse primary learning objectives to compensate for the lack of large-scale, real-world, trial-and-error data. While these auxiliary rewards accelerate learning, they require significant engineering effort, may introduce human biases, and cannot adapt to the robot’s evolving capabilities during training. In this paper, we introduce Reward Training Wheels (RTW), a teacher-student framework that automates auxiliary reward adaptation for robotics RL. To be specific, the RTW teacher dynamically adjusts auxiliary reward weights based on the student’s evolving capabilities to determine which auxiliary reward aspects require more or less emphasis to improve the primary objective. We demonstrate RTW on two challenging robot tasks: navigation in highly constrained spaces and off-road vehicle mobility on vertically challenging terrain. In simulation, RTW outperforms expert-designed rewards by 2.35% in navigation success rate and improves off-road mobility performance by 122.62%, while achieving 35% and 3X faster training efficiency, respectively. Physical robot experiments further validate RTW’s effectiveness, achieving a perfect success rate (5/5 trials vs. 2/5 for expert-designed rewards) and improving vehicle stability with up to 47.4% reduction in orientation angles.

## I. INTRODUCTION

Designing robust and sample-efficient Reinforcement Learning (RL) methods for real-world robotics remains an enduring challenge. Whether the goal is to enable a ground robot to navigate narrow corridors or to drive an off-road vehicle up steep, uneven terrain, reward functions are pivotal to guide an agent’s behavior. In principle, one could rely on sparse, goal-only rewards—for example, awarding a single bonus upon successfully reaching a target. However, such sparse feedback typically proves insufficient for complex robotic tasks, especially considering real-world data scarcity, as the agent has little incentive to avoid obstacles or maintain safe maneuvers, making it difficult to reach the goal and receive the sparse reward through trial and error [1], [2].

To address this limitation, roboticists introduce auxiliary rewards—terms that penalize collisions, encourage partial progress, or maintain safety constraints. While these auxiliary signals can dramatically accelerate learning, they also multiply the engineering burden: one must decide which auxiliary terms to include and how much to weigh each one, often through human trial and error and still leading to biased reward design [3]–[5]. Worse yet, an auxiliary reward that benefits early exploration (e.g., collision penalties to teach cautious movement) can become counterproductive later, as

the robot may overly prioritize always staying unnecessarily far from obstacles rather than completing the task efficiently.

Recent works in robotic RL reveal the extent of this auxiliary reward design burden. Tasks like navigation in highly constrained spaces [6]–[8] commonly include several hand-tuned rewards (e.g., distance-to-goal bonuses, collision penalties, velocity constraints, or time penalties), carefully balanced by human intuition [9]–[11]. Similarly, off-road mobility on vertically challenging terrain often relies on auxiliary terms to mitigate unsafe maneuvers—penalizing rollovers or slipping—while rewarding forward progress [12]. In each case, manually engineering these auxiliary signals can be time-consuming and prone to human biases and overfitting a particular training phase. These challenges highlight the need for a principled approach that can adapt auxiliary rewards throughout the training process.

Imagine training wheels used to learn how to ride a bicycle. An effective human teacher uses training wheels at the beginning to guide learning in a safe manner, encourages the student to depend less and less on them as learning progresses, and eventually completely remove them. Inspired by the diminishing role played by the training wheels, we hypothesize that RL auxiliary rewards also need to adapt to the student capabilities. In this paper, we introduce Reward Training Wheels (RTW), a teacher-student framework that preserves the robot’s ultimate task objective (learning to ride a bicycle) while dynamically adapting auxiliary components (reducing the reliance on the training wheels). The teacher agent monitors key training signals—such as success rate and other performance metrics—to assess the student robot’s current capability and adjust auxiliary reward weights accordingly. As a result, the student leverages RTW to improve learning efficiency and performance through the adaptive auxiliary rewards throughout the learning process.

We apply RTW on two difficult robotic tasks: (1) navigation in highly constrained spaces and (2) off-road vehicle mobility on vertically challenging terrain. Compared to expert-designed auxiliary rewards, our approach outperforms by 2.35% in navigation success rate and improves off-road mobility performance by 122.62%, while achieving 35% and 3X faster training efficiency, respectively. Physical robot experiments further validate RTW’s effectiveness, achieving a perfect success rate (5/5 trials vs. 2/5 for expert-designed rewards) and improving vehicle stability with up to 47.4% reduction in orientation angles.

All authors are with the Department of Computer Science, George Mason University {lwang44, txu25, ylu22, xiao}@gmu.edu

\*Equally contributing authors

## II. RELATED WORK

RTW is related with curriculum learning and reward shaping, for which we review related work in this section.

### A. Curriculum Learning

Curriculum learning, first formalized by Bengio et al. [13], has become increasingly important in RL as a strategy to improve sample efficiency and final performance. Traditional curriculum approaches in robotics focus on progressively increasing task difficulty [4], often by modifying environmental parameters such as obstacle density [14] or terrain complexity [15]. While effective, these approaches introduce additional engineering complexity by requiring multiple task variants and can lead to distribution shifts when transferring to the target environment. Unlike these methods that primarily modify task configurations or initial states, our teacher-student framework maintains a consistent environment while dynamically adapting the auxiliary reward structure as the student progresses.

### B. Reward Shaping

Reward shaping has long been recognized as a critical element in RL [1], [2]. Potential-based reward shaping [1] provides theoretical guarantees for preserving optimal policies, though how to design potential functions to efficiently lead to optimality remains challenging. Methods for automatic reward design have emerged as alternatives to manual engineering. Singh et al. [16] explored intrinsically motivated RL, where auxiliary rewards emerge from the agent's learning progress. More recently, Devidze et al. [5] proposed metrics to evaluate the informativeness of reward signals during training. These approaches recognize the limitations of fixed auxiliary rewards but typically do not adapt rewards based on the agent's current capabilities as RTW does.

Several recent studies have explored adaptive rewards in robotic RL. Gupta et al. [17], [18] investigated how different reward formulations affect robotic manipulation learning, showing that reward design choices significantly impact training dynamics. Riedmiller et al. [19] introduced a learning from play framework that separates reward specification from policy learning, allowing more natural skill acquisition. In the context of mobile robots, Xu et al. [10] benchmarked different reward structures for navigation tasks, while Xu et al. [20] explored reward design for off-road mobility. These studies highlight the challenge of creating effective reward functions for complex robotic tasks but typically rely on static reward formulations. Our approach builds upon these insights by introducing automatic adaptation of auxiliary rewards throughout the training process.

## III. APPROACH

RTW is a dual-agent framework for dynamic auxiliary reward design in robotics RL. As depicted in Fig. 1, our framework employs a hierarchical structure comprising two specialized agents:

- A Partially Observable Markov Decision Process (POMDP) for the student agent ( $\cdot^S$ ).

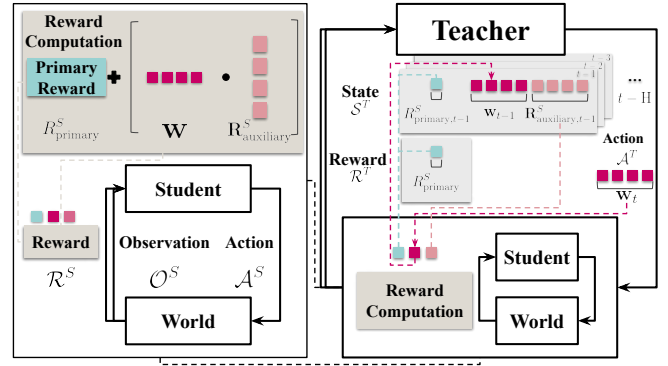


Fig. 1: Overview of Reward Training Wheel: (Left) The student agent interacts with the world and receives a reward composed of a primary component and weighted auxiliary components. (Right) The teacher agent maintains the history of previous weights, primary rewards, and auxiliary rewards as its state, and generates new weights as its action to optimize the student's learning process.

- A Markov Decision Process (MDP) for the teacher agent ( $\cdot^T$ ) generating auxiliary reward weights.

### A. Student Agent

The student agent operates in a POMDP defined as  $\mathcal{M}^S = \langle \mathcal{S}^S, \mathcal{A}^S, \mathcal{O}^S, \mathcal{T}^S, \Omega^S, \mathcal{R}^S, \gamma^S \rangle$ , where  $\mathcal{S}^S$  is the state space of the robotic task,  $\mathcal{A}^S$  is the robot action space, and  $\mathcal{O}^S$  is the robot observation space. The transition function  $\mathcal{T}^S : \mathcal{S}^S \times \mathcal{A}^S \rightarrow \mathcal{S}^S$  defines state dynamics, while the observation function  $\Omega^S : \mathcal{S}^S \times \mathcal{A}^S \rightarrow \mathcal{O}^S$  determines what the agent can perceive. The reward function  $\mathcal{R}^S : \mathcal{S}^S \times \mathcal{A}^S \times \mathcal{S}^S \times \mathbf{w} \rightarrow \mathbb{R}$  is parameterized by auxiliary reward weights  $\mathbf{w}$ , and  $\gamma^S \in [0, 1]$  is the discount factor. The student agent's goal is to learn a policy  $\pi^S : \mathcal{O}^S \rightarrow \mathcal{A}^S$  that maximizes the expected cumulative reward.

1) *Decomposed Reward Structure:* We decompose the student's reward function into a primary reward component and multiple auxiliary components:

$$\mathcal{R}^S(s, a, s', \mathbf{w}) = R_{\text{primary}}^S(s, a, s') + \sum_{k=1}^K w_k \cdot \mathbf{R}_{\text{aux},k}^S(s, a, s') \quad (1)$$

where:

- $R_{\text{primary}}^S(s, a, s')$  is the primary reward (e.g., successful task completion),
- $\mathbf{R}_{\text{aux},k}^S(s, a, s')$  is the  $k$ -th auxiliary reward component, and
- $w_k \in [0, 1]$  is the weight generated by the teacher agent

For each component, the teacher dynamically generates a weight  $w_k$  that determines the importance of that auxiliary reward during the learning progress.

### B. Teacher Agent

The teacher agent operates in a fully observable Markov Decision Process (MDP) defined as  $\mathcal{M}^T = \langle \mathcal{S}^T, \mathcal{A}^T, \mathcal{T}^T, \mathcal{R}^T, \gamma^T \rangle$ , where  $\mathcal{S}^T$  is the teacher state space containing the history of student performance and reward

information. The teacher's action space  $\mathcal{A}^T = [0, 1]^K$  represents the normalized auxiliary reward weights  $\mathbf{w}$ , with  $K$  being the number of auxiliary reward components. The transition function  $\mathcal{T}^T : \mathcal{S}^T \times \mathcal{A}^T \rightarrow \mathcal{S}^T$  governs state dynamics, while the reward function  $\mathcal{R}^T : \mathcal{S}^T \times \mathcal{A}^T \times \mathcal{S}^T \rightarrow \mathbb{R}$  is based on student task performance. The discount factor  $\gamma^T \in [0, 1]$  determines the importance of future rewards.

The teacher's state  $s_t^T \in \mathcal{S}^T$  at time  $t$  comprises three main components with a history horizon  $H$ :

$$s_t^T = \{\mathbf{w}_{t-H:t-1}, \mathbf{p}_{t-H:t-1}, \mathbf{r}_{t-H:t-1}\} \quad (2)$$

where:

- $\mathbf{w}_{t-H:t-1} = \{\mathbf{w}_i\}_{i=t-H}^{t-1}$  is the history of auxiliary reward weight configurations,
- $\mathbf{p}_{t-H:t-1} = \{p_i\}_{i=t-H}^{t-1}$  is the history of primary rewards, and
- $\mathbf{r}_{t-H:t-1} = \{\mathbf{r}_i^k\}_{i=t-H, k=1}^{t-1, K}$  is the history of auxiliary reward vectors.

This state representation allows the teacher to track three critical aspects of the student's learning process: (1) what weight configurations have been assigned previously, (2) how well the student performs overall on the primary objective, and (3) how the student performs with respect to each auxiliary reward component.

### C. Learning Process

1) *Student Learning*: The student agent learns to perform the task using RL (e.g., PPO [21]) in the environment to maximize the expected cumulative reward:

$$J^S(\pi_{\theta^S}^S, \mathbf{w}) = \mathbb{E}_{\tau^S \sim \pi_{\theta^S}^S} \left[ \sum_{t=0}^T (\gamma^S)^t \mathcal{R}^S(s_t, a_t, s_{t+1}, \mathbf{w}_t) \right] \quad (3)$$

where  $\tau^S = (s_0, a_0, s_1, a_1, \dots, s_T)$  is a trajectory sampled from the student policy  $\pi_{\theta^S}^S$ , parameterized by  $\theta^S$ . At each training iteration, the student collects trajectories using the current policy and reward weights and then updates its policy parameters:

$$\theta^S \leftarrow \theta^S + \eta^S \nabla_{\theta^S} J^S(\pi_{\theta^S}^S, \mathbf{w}) \quad (4)$$

where  $\eta^S$  is the learning rate.

2) *Teacher Learning*: The teacher agent monitors the student's performance metrics to design adaptive auxiliary reward structures. The student performance history combined with reward history reveals what specific aspects of the task the student struggles with and masters. Based on this assessment, the teacher generates appropriate auxiliary reward weights  $\mathbf{w}_t$  that emphasize aspects that need improvement while reducing emphasis on what has already been mastered.

The teacher's objective is to maximize the student's task performance, which we define as:

$$J^T(\pi_{\theta^T}^T; \mathbf{w}) = \mathbb{E}_{\tau^T \sim \pi_{\theta^T}^T} \left[ \sum_{t=0}^T \gamma_t^T (r_t^T; \mathbf{w}_t) \right] \quad (5)$$

where  $\tau^T = (s_0^T, a_0^T, s_1^T, a_1^T, \dots, s_T^T)$  is a trajectory in the teacher's MDP, and  $r_t^T$  is the teacher's reward at time  $t$ ,

### Algorithm 1 Reward Training Wheels

- 1: **Input**: Initial parameters  $\theta^S$  and  $\theta^T$ , learning rates  $\eta^S$  and  $\eta^T$ , primary reward  $R_{\text{primary}}^S$ , auxiliary reward components  $\{\mathbf{R}_{\text{aux},k}^S\}_{k=1}^K$ , history horizon  $H$ , number of iterations  $N$
- 2: **Output**: Trained student policy  $\pi_{\theta^S}^S$
- 3: Initialize student policy  $\pi_{\theta^S}^S$ , teacher policy  $\pi_{\theta^T}^T$ , and teacher state  $s_0^T$
- 4: **for**  $t = 0$  to  $N - 1$  **do**
- 5:   Generate auxiliary reward weights  $\mathbf{w}_t = \pi_{\theta^T}^T(s_t^T)$
- 6:   Formulate student reward  $\mathcal{R}^S = R_{\text{primary}}^S + \sum_{k=1}^K w_{t,k} \cdot \mathbf{R}_{\text{aux},k}^S$
- 7:   Collect student trajectories  $\{\tau_i^S\}_{i=1}^M$  using current policy and reward function
- 8:   Compute performance metrics  $\mathbf{p}_t$  and auxiliary reward values  $\mathbf{r}_t$
- 9:   Update student policy:  $\theta^S \leftarrow \theta^S + \eta^S \nabla_{\theta^S} J^S(\pi_{\theta^S}^S, \mathbf{w})$
- 10:   Update teacher state with history horizon:  $s_{t+1}^T = \{\mathbf{w}_{t-H+1:t}, \mathbf{p}_{t-H+1:t}, \mathbf{r}_{t-H+1:t}\}$
- 11:   Update teacher policy:  $\theta^T \leftarrow \theta^T + \eta^T \nabla_{\theta^T} J^T(\pi_{\theta^T}^T; \mathbf{w})$
- 12: **end for**
- 13: **return**  $\pi_{\theta^S}^S$

defined as the student's primary reward under the current auxiliary reward weights  $\mathbf{w}_t$ .

We implement the teacher using a separate RL agent that observes the student's performance metrics and outputs auxiliary reward weights. The teacher's policy is updated using:

$$\theta^T \leftarrow \theta^T + \eta^T \nabla_{\theta^T} J^T(\pi_{\theta^T}^T; \mathbf{w}) \quad (6)$$

where  $\eta^T$  is the teacher's learning rate.

### D. Training Algorithm

Algorithm 1 summarizes our training procedure. We initialize both student and teacher policies (line 3). For each training iteration, the teacher generates auxiliary reward weights based on its current state (line 5), which are used to form the student's reward function (line 6). The student collects trajectories using this reward function (line 7), and performance metrics are calculated (line 8). The student policy is then updated (line 9), and the teacher's state is updated to include the latest weight configuration and performance metrics (line 10). Finally, the teacher policy is updated based on the student's performance (line 11). This process continues until convergence.

## IV. EXPERIMENTS AND RESULTS

In this section, we evaluate RTW's effectiveness in dynamic auxiliary reward design by comparing it against three baselines: Expert-designed Reward (ER), which uses fixed auxiliary weights; Reward Randomization (RR), where auxiliary weights change randomly; and Model Predictive Path Integral (MPPI) [22], a conventional planner. We test on two challenging robotic domains: Confined-Space Navigation and Off-Road Vehicle Mobility (Fig. 3). We also conduct physical experiments to validate our approach in real-world settings. Here, we describe our experimental setup, evaluation metrics, and results.

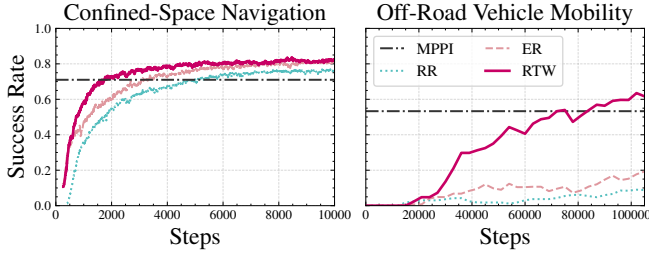


Fig. 2: Learning curves showing success rate vs. training steps for MPPI, RR, ER, and RTW. Left: Confined-Space Navigation task. Right: Off-Road Vehicle Mobility task. RTW achieves higher success rate with fewer training steps in both tasks.

### A. Simulation Experiments

1) *Experimental Setup*: We conduct simulation experiments in two domains: (1) Confined-Space Navigation and (2) Off-Road Vehicle Mobility.

a) *Confined-Space Navigation*: The environment consists of randomly generated maps with narrow corridors and obstacles. The robot navigates from a starting position to a target location without collisions. The reward components include:

- $\mathcal{R}_{\text{primary}}^S$ : +20.0 for reaching the target,
- $\mathcal{R}_{\text{aux},1}^S$ : -2.0 for collisions with obstacles,
- $\mathcal{R}_{\text{aux},2}^S$ : +2.0 for progress distance to the goal, and
- $\mathcal{R}_{\text{aux},3}^S$ : +1.0 for maintain safe distance from obstacles.

b) *Off-Road Vehicle Mobility*: The environment features procedurally generated terrain with varying slopes, roughness, and obstacles. The vehicle must reach a target while maintaining stability. The reward components include:

- $\mathcal{R}_{\text{primary}}^S$ : +1000.0 for reaching the target,
- $\mathcal{R}_{\text{aux},1}^S$ : +5.0 for progress distance to the goal,
- $\mathcal{R}_{\text{aux},2}^S$ : +1.0 for vehicle current speed (up to maximum speed - 4m/s),
- $\mathcal{R}_{\text{aux},3}^S$ : -0.5 for vehicle stalling or lack of progress, and
- $\mathcal{R}_{\text{aux},4}^S$ : -0.5 for exceeding roll and pitch threshold angles.

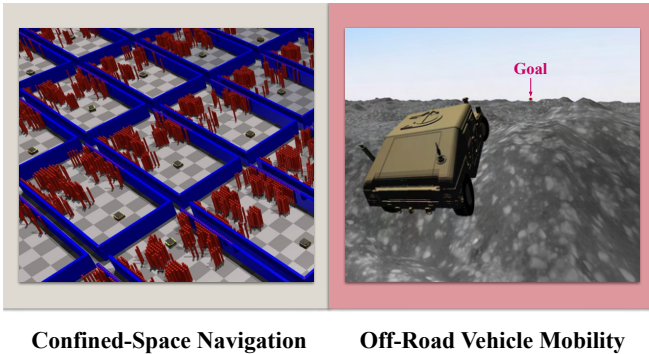


Fig. 3: Simulation environments for the two robotics tasks: (Left) Confined-Space Navigation, requiring precise maneuvering through narrow corridors. (Right) Off-Road Vehicle Mobility, challenging the agent to traverse uneven terrain to reach a goal.

TABLE I: Hyperparameters for RTW Implementation.

|                        | Navigation | Mobility |
|------------------------|------------|----------|
| <b>Teacher</b>         |            |          |
| Learning Rate          | 3e-4       | 3e-4     |
| PPO Epochs             | 10         | 10       |
| Size of State Space    | 7          | 9        |
| Size of Action Space   | 3          | 4        |
| Episodes per Update    | 10         | 2        |
| <b>Student</b>         |            |          |
| Learning Rate          | 3e-4       | 5e-4     |
| PPO Epochs             | 10         | 5        |
| Batch Size             | 64         | 1500     |
| Timesteps/iter         | 256        | 3000     |
| Policy Network         | [512, 512] | [64, 64] |
| Value Network          | [512, 512] | [64, 64] |
| Number of Environments | 128        | 5        |

c) *Implementation Details*: We implement RTW and the RL baselines using PPO with the hyperparameters in Table I. For MPPI, we use standard parameters with a planning horizon of 2.0 seconds.

d) *Evaluation Metrics*: For Confined-Space Navigation, we measure: (i) Success Rate (percentage of trials reaching goal without collisions), (ii) Mean Traversal Time (seconds per successful trial), (iii) Average Distance to Goal (final distance in meters for all trials), and (iv) Average Speed (meters per second, with a maximum of 2 m/s).

For Off-Road Vehicle Mobility, we track: (i) Success Rate out of 30 trials (percentage of reaching goal without rollover or getting stuck), (ii) Mean traversal time (of successful trials in seconds), (iii) Average roll/pitch angles with variance (in degrees).

### 2) Simulation Results:

a) *Performance Comparison*: Table II presents quantitative results across both domains. RTW consistently outperforms all baselines in every metric.

In the Confined-Space Navigation task, RTW achieves the highest success rate among all approaches (82.67%). MPPI shows the lowest success rate, highlighting the difficulty of this navigation task for conventional planning approaches. ER performs reasonably well with expert-designed weights, while RR's inferior performance demonstrates that merely randomizing auxiliary reward weights is detrimental. For Off-Road Vehicle Mobility, RTW achieves a 76.67% success rate, outperforming the next best MPPI approach. This represents a 43.76% relative improvement over the conventional planning approach and a 122.62% improvement over the ER baseline. RTW also achieves the fastest traversal time, indicating its ability to navigate efficiently through vertically challenging terrain. While MPPI and RR show slightly better absolute roll and pitch metrics, respectively, these come at the cost of substantially lower success rates, suggesting overly conservative navigation strategies. RTW maintains reasonable vehicle stability metrics while successfully completing far more trajectories, demonstrating its ability to balance multiple competing objectives in complex off-road environments.



TABLE II: Performance Comparison on Confined-Space Navigation and Off-Road Vehicle Mobility (mean  $\pm$  std).  $\uparrow$  indicates higher is better;  $\downarrow$  indicates lower is better.

| Method            | Confined-Space Navigation |                        |                            |                        | Off-Road Vehicle Mobility |                         |                            |                             |
|-------------------|---------------------------|------------------------|----------------------------|------------------------|---------------------------|-------------------------|----------------------------|-----------------------------|
|                   | Success $\uparrow$        | Time $\downarrow$      | Dist. to Goal $\downarrow$ | Avg. Speed $\uparrow$  | Success $\uparrow$        | Time $\downarrow$       | Absolute Roll $\downarrow$ | Absolute Pitch $\downarrow$ |
| ER                | 80.32 $\pm$ 1.98          | 6.23 $\pm$ 0.65        | 1.52 $\pm$ 0.19            | 1.24 $\pm$ 0.43        | 34.44 $\pm$ 9.56          | 31.68 $\pm$ 0.83        | 5.47 $\pm$ 0.08            | 5.33 $\pm$ 0.06             |
| RR                | 76.15 $\pm$ 2.74          | 6.89 $\pm$ 0.97        | 1.76 $\pm$ 0.37            | 1.12 $\pm$ 0.31        | 12.22 $\pm$ 6.85          | 30.43 $\pm$ 1.25        | 5.16 $\pm$ 0.35            | <b>5.12</b> $\pm$ 0.13      |
| MPPI              | 71.01 $\pm$ 2.21          | 6.10 $\pm$ 1.02        | 1.89 $\pm$ 0.47            | 0.91 $\pm$ 0.36        | 53.33 $\pm$ 2.03          | 30.26 $\pm$ 1.54        | <b>5.10</b> $\pm$ 16.00    | 6.51 $\pm$ 35.39            |
| <b>RTW (Ours)</b> | <b>82.67</b> $\pm$ 1.95   | <b>6.02</b> $\pm$ 0.86 | <b>1.37</b> $\pm$ 0.17     | <b>1.32</b> $\pm$ 0.28 | <b>76.67</b> $\pm$ 2.72   | <b>29.39</b> $\pm$ 0.56 | 5.52 $\pm$ 0.17            | 6.06 $\pm$ 0.37             |

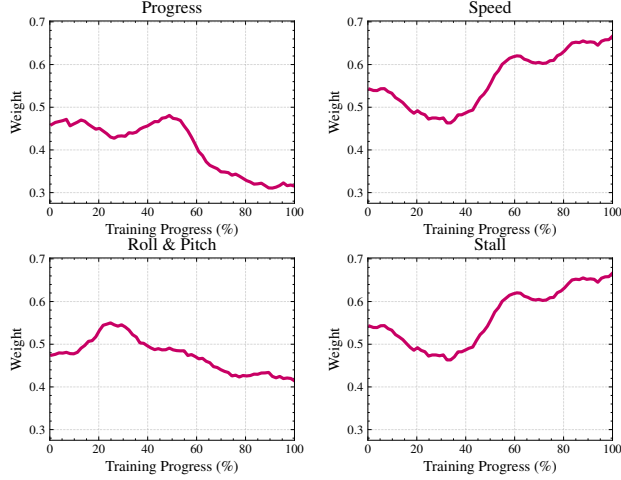


Fig. 4: Evolution of auxiliary reward weights during training in Off-Road Vehicle Mobility.

*b) Training Efficiency:* Fig. 2 presents the learning curves for all methods. RTW consistently achieves high performance with fewer training steps than the RL baselines. In the navigation task, RTW reaches an 80% success rate approximately 35% faster than ER, while RR fails to reach this threshold. In the off-road mobility task, RTW demonstrates sample efficiency, reaching the same performance threshold approximately 3X faster than ER. Neither ER nor RR achieves comparable final performance to RTW, with both methods showing maximum success rates of only about 20% even after completing the full training process.

*c) Auxiliary Weights Evolution:* Figure 4 illustrates how RTW dynamically adapts auxiliary reward weights throughout training in the Off-Road Mobility task. The progress reward weight decreases from 0.45 to 0.32 in later stages, indicating reduced emphasis on simple forward movement once basic navigation is mastered. The roll-pitch penalty weight follows an inverted U-shaped pattern, first increasing to emphasize stability during skill acquisition, then decreasing as the vehicle develops robust control strategies. Both speed and stall prevention weights show significant increases (0.52 to 0.65) in the latter half of training, reflecting a shift toward performance optimization.

Notably, speed and stall prevention weights follow remarkably similar trajectories despite being separate components, suggesting RTW has discovered their intrinsic relationship—both encourage forward momentum. This demonstrates RTW’s ability to identify underlying relationships

TABLE III: Experiment Results of RTW, ER and RR: success rate, mean traversal time (of successful trials), and mean absolute roll and pitch angles.

|                             | RTW                                       | ER                                 | RR                                 |
|-----------------------------|---|------------------------------------|------------------------------------|
| Success Rate $\uparrow$     | <b>5/5</b>                                | 2/5                                | 1/5                                |
| Traversal Time $\downarrow$ | <b>6.05</b> $\pm$ 0.95                    | 6.48 $\pm$ 0.58                    | 8.68 $\pm$ 2.88                    |
| Absolute Roll $\downarrow$  | <b>4.8</b> $^{\circ}\pm$ 5.01 $^{\circ}$  | 9.12 $^{\circ}\pm$ 7.35 $^{\circ}$ | 7.82 $^{\circ}\pm$ 9.61 $^{\circ}$ |
| Absolute Pitch $\downarrow$ | <b>7.09</b> $^{\circ}\pm$ 3.71 $^{\circ}$ | 7.33 $^{\circ}\pm$ 4.91 $^{\circ}$ | 8.99 $^{\circ}\pm$ 7.70 $^{\circ}$ |

between reward components without explicit programming.

### B. Physical Demonstration

To validate the effectiveness of our framework, we deploy the RL policies, i.e., RTW, ER and RR learned in simulation, on a physical 1/10th scale open-source Verti-4-Wheeler robot [23] platform on an off-road mobility testbed constructed by rocks (Fig. 5). The robot is a four-wheeled platform based on an off-the-shelf, two-axle, four-wheel-drive, off-road vehicle from Traxxas. The onboard computation platform is a NVIDIA Jetson Xavier NX module. The results are presented in Table III.



Fig. 5: Physical Off-Road Testbed for RTW.

The physical validation results presented in Table III demonstrate the clear superiority of the RTW approach compared to ER and RR across all performance metrics. RTW achieves a perfect success rate (5/5 trials), substantially outperforming both ER (2/5) and RR (1/5).

RTW achieves the shortest mean traversal time compared to ER and RR. RTW demonstrates improved vehicle stability through significantly reduced orientation angles. For absolute roll, RTW achieves 4.80 $^{\circ}$  compared to ER’s 9.12 $^{\circ}$  (47.4% reduction) and RR’s 7.82 $^{\circ}$  (38.6% reduction). For absolute pitch, RTW’s 7.09 $^{\circ}$  outperforms ER’s 7.33 $^{\circ}$  (3.3% reduction) and RR’s 8.99 $^{\circ}$  (21.1% reduction).

## V. CONCLUSIONS AND FUTURE WORK

In this paper, we address the persistent challenge of designing effective auxiliary rewards for RL in complex robotics tasks. The reliance on manual reward engineering is time-consuming, prone to bias, and often fails to adapt to the robot's changing capabilities during training. To overcome these limitations, we introduce RTW, a novel teacher-student framework that automatically adapts auxiliary reward weights throughout the learning process. Through extensive simulation and physical experiments in challenging confined-space navigation and off-road vehicle mobility tasks, we demonstrate that RTW outperforms expert-designed rewards, reward randomization, and the classical MPPI planner. Notably, RTW demonstrated an interesting capability to identify functional relationships between auxiliary rewards, automatically synchronizing conceptually related components like speed and stall prevention without explicit programming.

Future work can focus on extending RTW to automatically discover and incorporate new auxiliary reward components, potentially leveraging techniques from unsupervised learning or intrinsic motivation. Investigating more sophisticated teacher agent architectures, such as those incorporating attention mechanisms or recurrent networks, can further improve the teacher's ability to model long-term dependencies in the student's learning process. Exploring the application of RTW to a broader range of robotic tasks, including manipulation and human-robot interaction, is also a promising direction.

## ACKNOWLEDGMENTS

This work has taken place in the RobotiXX Laboratory at George Mason University. RobotiXX research is supported by National Science Foundation (NSF, 2350352), Army Research Office (ARO, W911NF2320004, W911NF2420027, W911NF2520011), Air Force Research Laboratory (AFRL), US Air Forces Central (AFCENT), Google DeepMind (GDM), Clearpath Robotics, Raytheon Technologies (RTX), Tangenta, Mason Innovation Exchange (MIX), and Walmart.

## REFERENCES

- [1] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *ICML*, vol. 99. Citeseer, 1999, pp. 278–287.
- [2] D. Hadfield-Menell, S. Milli, P. Abbeel, S. J. Russell, and A. Dragan, "Inverse reward design," *Advances in neural information processing systems*, vol. 30, 2017.
- [3] Y. Lu and E. Plaku, "Leveraging single-goal predictions to improve the efficiency of multi-goal motion planning with dynamics," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 850–857.
- [4] L. Wang, Z. Xu, P. Stone, and X. Xiao, "Grounded curriculum learning," *arXiv preprint arXiv:2409.19816*, 2024.
- [5] R. Devidze, P. Kamalaruban, and A. Singla, "Informativeness of reward functions in reinforcement learning," *arXiv preprint arXiv:2402.07019*, 2024.
- [6] X. Xiao, Z. Xu, Z. Wang, Y. Song, G. Warnell, P. Stone, T. Zhang, S. Ravi, G. Wang, H. Karnan *et al.*, "Autonomous ground navigation in highly constrained spaces: Lessons learned from the benchmark autonomous robot navigation challenge at icra 2022 [competitions]," *IEEE Robotics & Automation Magazine*, vol. 29, no. 4, pp. 148–156, 2022.
- [7] X. Xiao, Z. Xu, G. Warnell, P. Stone, F. G. Guinjoan, R. T. Rodrigues, H. Bruyninckx, H. Mandala, G. Christmann, J. L. Blanco-Claraco *et al.*, "Autonomous ground navigation in highly constrained spaces: Lessons learned from the second barn challenge at icra 2023 [competitions]," *IEEE Robotics & Automation Magazine*, vol. 30, no. 4, pp. 91–97, 2023.
- [8] X. Xiao, Z. Xu, A. Datar, G. Warnell, P. Stone, J. J. Damanik, J. Jung, C. A. Deresa, T. D. Huy, C. Jinyu *et al.*, "Autonomous ground navigation in highly constrained spaces: Lessons learned from the third barn challenge at icra 2024 [competitions]," *IEEE Robotics & Automation Magazine*, vol. 31, no. 3, pp. 197–204, 2024.
- [9] D. Perille, A. Truong, X. Xiao, and P. Stone, "Benchmarking metric ground navigation," in *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2020, pp. 116–121.
- [10] Z. Xu, B. Liu, X. Xiao, A. Nair, and P. Stone, "Benchmarking reinforcement learning techniques for autonomous navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9224–9230.
- [11] Z. Xu, X. Xiao, G. Warnell, A. Nair, and P. Stone, "Machine learning methods for local motion planning: A study of end-to-end vs. parameter learning," in *2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2021, pp. 217–222.
- [12] T. Xu, C. Pan, and X. Xiao, "Reinforcement learning for wheeled mobility on vertically challenging terrain," in *2024 IEEE International Symposium on Safety Security Rescue Robotics (SSRR)*. IEEE, 2024, pp. 125–130.
- [13] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.
- [14] B. Ivanovic, J. Harrison, A. Sharma, M. Chen, and M. Pavone, "Barc: Backward reachability curriculum for robotic reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 15–21.
- [15] 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.
- [16] S. Singh, R. L. Lewis, A. G. Barto, and J. Sorg, "Intrinsically motivated reinforcement learning: An evolutionary perspective," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 2, pp. 70–82, 2010.
- [17] A. Gupta, A. Pacchiano, Y. Zhai, S. Kakade, and S. Levine, "Unpacking reward shaping: Understanding the benefits of reward engineering on sample complexity," *Advances in Neural Information Processing Systems*, vol. 35, pp. 15 281–15 295, 2022.
- [18] C. Zhao, Y. Li, Y. Jian, J. Xu, L. Wang, Y. Ma, and X. Jin, "Ti-nvm: Enhancing map accuracy and consistency with normal vector-assisted mapping," *IEEE Robotics and Automation Letters*, 2025.
- [19] M. Riedmiller, R. Hafner, T. Lampe, M. Neunert, J. Degraeve, T. Wiele, V. Mnih, N. Heess, and J. T. Springenberg, "Learning by playing solving sparse reward tasks from scratch," in *International conference on machine learning*. PMLR, 2018, pp. 4344–4353.
- [20] T. Xu, C. Pan, and X. Xiao, "Verti-selector: Automatic curriculum learning for wheeled mobility on vertically challenging terrain," *arXiv preprint arXiv:2409.17469*, 2024.
- [21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [22] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control: From theory to parallel computation," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 344–357, 2017.
- [23] A. Datar, C. Pan, M. Nazeri, and X. Xiao, "Toward wheeled mobility on vertically challenging terrain: Platforms, datasets, and algorithms," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16 322–16 329.