GACL: Grounded Adaptive Curriculum Learning with Active Task and Performance Monitoring

Linji Wang[1], Zifan Xu[2], Peter Stone[2,3], and Xuesu Xiao[1]

***Abstract*— Curriculum learning has emerged as a promising approach for training complex robotics tasks, yet current applications predominantly rely on manually designed curricula, which demand significant engineering effort and can suffer from subjective and suboptimal human design choices. While automated curriculum learning has shown success in simple domains like grid worlds and games where task distributions can be easily specified, robotics tasks present unique challenges: they require handling complex task spaces while maintaining relevance to target domain distributions that are only partially known through limited samples. To this end, we propose Grounded Adaptive Curriculum Learning (GACL[1]), a framework specifically designed for robotics curriculum learning with three key innovations: (1) a task representation that consistently handles complex robot task design, (2) an active performance tracking mechanism that allows adaptive curriculum generation appropriate for the robot's current capabilities, and (3) a grounding approach that maintains target domain relevance through alternating sampling between reference and synthetic tasks. We validate GACL on wheeled navigation in constrained environments and quadruped locomotion in challenging 3D confined spaces, achieving 6.8% and 6.1% higher success rates, respectively, than state-of-the-art methods in each domain.**

I. Introduction

Curriculum learning has shown promises in training robots for complex tasks such as navigating through highly constrained environments or maintaining quadruped locomotion across challenging terrain [1], [2]. However, current applications of curriculum learning in robotics face a fundamental challenge: they predominantly rely on manually designed curricula, which demand significant engineering effort and can suffer from subjective, suboptimal design choices. For example, in quadruped locomotion tasks [2], roboticists must carefully design progressive stages from basic jumping skills to complex obstacle traversal and manually define success metrics and progression conditions at each stage.

While automated curriculum learning (ACL) can reduce manual design effort, current approaches have primarily focused on simple domains like grid worlds and games where tasks can be easily specified [3]. Recent advances like PAIRED [4] and CLUTR [5] have improved upon basic ACL by introducing teacher agents for task generation. However, these methods cannot be directly applied to complex robotics problems, since their teacher agents use simplified observation spaces that lack critical information about task complexity and student learning progress. More fundamentally, robotics tasks present a unique challenge overlooked by existing ACL approaches: open-ended curricula often move towards tasks that are not relevant to target deployment domains. Every robotics problem has a final deployment domain (like household environments for home robots) whose complete distribution is unknown and can only be partially observed through limited samples. While generative models can approximate these distributions, robotic curriculum learning requires generating tasks that not only match the target distribution but also adapt to the robot's current learning capabilities. This creates a critical balance: the curriculum must generate diverse training scenarios that challenge the robot during learning while ensuring these scenarios remain relevant to the target domain where the robot will eventually operate.

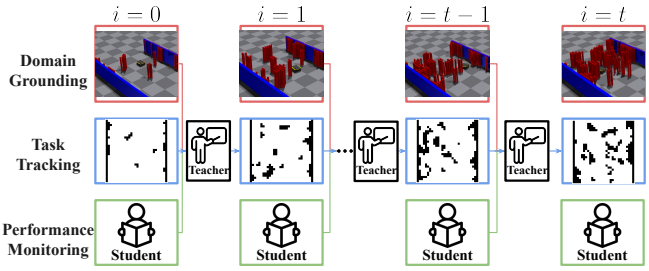

Fig. 1: GACL creates an adaptive sequence of robot tasks with progressively increasing difficulty through target domain grounding, task tracking, and performance monitoring.

To address these challenges, we propose Grounded Adaptive Curriculum Learning (GACL), a framework specifically designed for robotics curriculum learning. This paper introduces three key innovations: (1) a task representation that consistently handles complex robot task design, (2) an active performance tracking mechanism that allows adaptive curriculum generation appropriate for the robot's current capabilities, and (3) a grounding approach that maintains target domain relevance through alternating sampling between reference and synthetic tasks (Fig. 1).

We validate GACL on two challenging robotics domains: wheeled robot navigation in highly constrained environments [6] and quadruped locomotion in confined 3D spaces [7]. GACL demonstrates consistent improvements in both domains, achieving 6.8% and 6.1% higher success rates, respectively, compared to state-of-the-art curriculum learning methods.

II. Related Work

Curriculum learning, originally inspired by the way humans acquire skills, has been explored in robotics to structure

[1]Department of Computer Science, George Mason University {lwang44, xiao}@gmu.edu [2]Department of Computer Science, The University of Texas at Austin zfxu@utexas.edu, pstone@cs.utexas.edu [3]Sony AI

[1]https://github.com/linjiw/GACL

learning from increasingly complex tasks [8]. Early work often relied on carefully hand-crafted curricula, where experts define stage-by-stage task progression, such as denser obstacle distributions [9], increased elevation changes [10], and more complex locomotion strategies [2]. While these manual curricula can improve learning, they require significant domain expertise, often introducing subjective biases that can lead to suboptimal training efficiency and final performance [11]. To address these limitations, researchers have turned to ACL to reduce human effort and automatically adapt training to improve efficiency and performance.

ACL seeks to offload the burden of manually designing a sequence of tasks to a teacher agent that observes the student agent's learning progress and adaptively proposes new tasks [3]. A primary goal in ACL is to challenge the student agent appropriately—neither too easy nor too hard—thereby improving sample efficiency and robustness. However, most ACL work were developed in relatively simple domains, such as grid worlds and basic physics simulations [12], where tasks can be represented in low-dimensional spaces.

Recent studies on Unsupervised Environment Design (UED) highlight more advanced teacher-student frameworks. One prominent example is PAIRED [4], which uses a regret-based objective to push the student agent toward increasingly challenging environments. CLUTR [5] builds upon PAIRED by integrating a learned latent representation (using variational autoencoders [13]) for task generation, offering more efficient task sampling. Despite these advances, existing UED methods still exhibit notable limitations when applied to complex robotics tasks: (1) The task space remains relatively simple in CLUTR, which focuses on low-dimensional representations. In contrast, real-world robotics tasks often require more complex inputs—such as multi-channel images for navigation maps or multi-layer 3D terrain for legged locomotion—that cannot be easily captured by low-dimensional encoding; and (2) Most teacher implementations, including CLUTR's, operate in a stateless, multi-armed bandit fashion. This stateless approach restricts the teacher's capacity to model the nuanced relation between student performance and task complexity, which is particularly problematic for robotics scenarios involving high-dimensional sensor data and intricate physical dynamics.

One of the most critical yet underexplored challenges in applying ACL to robotics is ensuring that the automatically generated tasks remain relevant to real-world deployment scenarios. While open-ended task generation can foster broader policy generalization, unconstrained approaches risk producing training scenarios that deviate significantly from the environments in which the robot will operate. This mismatch can lead to wasted training effort on unrealistic tasks and failed task execution when facing real-world settings. Therefore, grounding the curriculum to the robot's target deployment domain with existing real-world experiences is therefore crucial. Yet, most existing ACL methods, including PAIRED and CLUTR, do not balance the need for realistic task distributions and pushing the agent's capabilities.

In summary, existing curriculum learning solutions face three central challenges in robotics: (1) scaling automated curricula to high-dimensional, complex tasks; (2) monitoring student performance to adapt tasks efficiently; and (3) preserving domain relevance while still pushing the agent's capabilities. In the next section, we introduce GACL, a framework designed explicitly to overcome these hurdles by combining richer task representations, active performance monitoring, and a grounding mechanism that ensures each generated task reflects real-world deployment conditions.

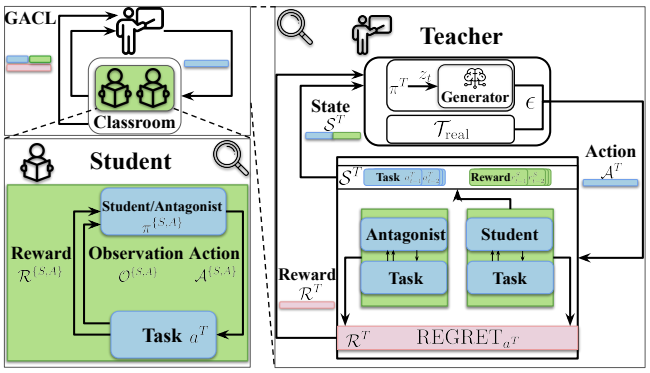

Fig. 2: Overview of the Dual-Agent GACL Framework (top left): student POMDP (bottom left) and teacher MDP (right).

III. Approach

We propose GACL, a dual-agent framework designed for adaptive curriculum learning in complex robotic tasks while maintaining target domain relevance. As depicted in Fig. 2, GACL involves two interacting components:

- A Partially Observable Markov Decision Process (POMDP) for the student agent ($\cdot^S$) learning the task.
- A fully informed Markov Decision Process (MDP) for the teacher agent ($\cdot^T$) generating a curriculum of tasks.

A. Dual-Agent (PO)MDP

1) Student Agent POMDP: The student agent operates in a POMDP defined as a tuple $\mathcal{M}^S = \langle \mathcal{S}^S, \mathcal{A}^S, \mathcal{O}^S, \mathcal{T}^S, \Omega^S, \mathcal{R}^S, \gamma^S \rangle$, where:

- $\mathcal{S}^S$ is the state space of the robotic task,
- $\mathcal{A}^S$ is the robot action space,
- $\mathcal{O}^S$ is the robot observation space,
- $\mathcal{T}^S : \mathcal{S}^S \times \mathcal{A}^S \to \mathcal{S}^S$ is the POMDP transition function,
- $\Omega^S : \mathcal{S}^S \times \mathcal{A}^S \to \mathcal{O}^S$ is the robot observation function,
- $\mathcal{R}^S : \mathcal{S}^S \times \mathcal{A}^S \times \mathcal{S}^S \to \mathbb{R}$ is the robot reward function based on task execution performance, and
- $\gamma^S \in [0, 1]$ is the student POMDP's discount factor.

The Student agent's goal is to learn a policy $\pi^S : \mathcal{O} \to \mathcal{A}$ that maximizes the expected cumulative reward in the partially observable task environment generated by the teacher.

2) Teacher Agent MDP: In contrast to the student's POMDP, the teacher agent operates in an 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, consisting of the comprehensive history of tasks and student performances,
- $\mathcal{A}^T$ is the teacher action space representing all possible tasks that can be assigned to the student,

- $\mathcal{T}^T : \mathcal{S}^T \times \mathcal{A}^T \to \mathcal{S}^T$ is the MDP transition function,
- $\mathcal{R}^T : \mathcal{S}^T \times \mathcal{A}^T \times \mathcal{S}^T \to \mathbb{R}$ is the teacher reward function based on student performance, and
- $\gamma^T \in [0, 1]$ is the discount factor for the teacher's MDP.

$s_t^T \in \mathcal{S}^T$ at time t is defined as $s_t^T = \{(a_i^T, r_i^S)\}_{i=0}^{t-1}$, where $a_i^T \in \mathcal{A}^T$ is the i th task assigned by the teacher and r_i^S is the student's performance (reward) for task i.

B. Grounded Adaptive Curriculum Learning (GACL)

GACL employs a hierarchical structure where a fully informed teacher agent guides the learning of a student agent, resembling a classroom setting (Fig. 2, top left). In this metaphor, the teacher (right side of the figure) not only determines the curriculum (i.e., the tasks) but also monitors the student's progress and compares it against an antagonist agent (bottom left). This design mirrors real-world educational scenarios, where teachers have comprehensive knowledge of both course material and student performance. Leveraging this vantage point, GACL introduces three key innovations for complex robotics tasks: *a latent generative model* that consistently encodes and reconstructs high-dimensional task environments, *active performance tracking* to adaptively challenge the student at its evolving skill level, and *domain relevance maintenance* via alternating between synthetic tasks and limited real-world references. Together, these elements ensure that GACL provides diverse yet realistic training scenarios, prevents the learning process from drifting into irrelevant tasks, and aligns the curriculum with the student's incremental improvements.

1) Task Representation via Latent Generative Model: We employ a Variational Autoencoder (VAE) to learn a continuous latent space $\mathcal{Z}$ for high-dimensional robotic tasks, trained on a partially known real-world task set $\mathcal{T}_{\text{real}}$. The encoder-decoder architecture compresses and reconstructs complex environments (e.g., 2D navigation maps or 3D terrain heightmaps). During curriculum learning, the teacher agent controls task generation by sampling latent vectors $\mathcal{Z}$, which the VAE decoder maps to concrete tasks. This VAE is pre-trained offline on $\mathcal{T}_{\text{real}}$ prior to the main GACL training loop, providing a stable latent representation throughout the curriculum learning process.

2) Student and Antagonist Agents: The student agent learns to perform a task using a reinforcement learning algorithm (e.g., PPO [14]) in the partially observable environment generated by the teacher. Its objective is to maximize the expected cumulative reward:

$$J^S(\pi_{\theta^S}^S) = \mathbb{E}_{\tau^S \sim \pi_{\theta^S}^S} \left[\sum_{t=0}^{T} \left(\gamma^S\right)^t r_t^S \right], \tag{1}$$

where τ^S is a trajectory sampled from the student policy $\pi_{\theta^S}^S$, parameterized by θ^S. To guide curriculum generation and evaluate the student's progress, we introduce an antagonist agent, following the flexible regret setting from PAIRED [4]. The antagonist is trained with the same observability and hyperparameters as the student, sharing the same objective function (Eqn. (1)), with J^A and $\pi_{\theta_A}^A$ as the antagonist's objective and policy respectively.

Algorithm 1 Grounded Adaptive Curriculum Learning

Input: VAE decoder G, initial parameters θ^S, θ^A, and θ^T, learning rates η^S, η^A, and η^T, real-world task set $\mathcal{T}_{\text{real}}$, and grounding probability ϵ
Output: Trained policies $\pi_{\theta^S}^S$, $\pi_{\theta^A}^A$, and $\pi_{\theta^T}^T$
Pretrain G with available real-world tasks $\mathcal{T}_{\text{real}}$
Initialize $\pi_{\theta^S}^S$, $\pi_{\theta^A}^A$, $\pi_{\theta^T}^T$, and $s_0^T = \{\}$
$t \leftarrow 0$
while not converged **do**
  $a_t^T \leftarrow \begin{cases} \text{sample from } \mathcal{T}_{\text{real}}, & \text{with probability } \epsilon, \\ \pi_{\theta^T}^T(s_t^T), & \text{with probability } 1-\epsilon, \end{cases}$
  Collect student trajectory in task a_t^T and compute cumulative reward $r_{a_t^T}^S$
  Collect antagonist trajectory in task a_t^T
  Compute regret $\textsc{Regret}_{a_t^T} \leftarrow V_{a_t^T}(\pi_{\theta^A}^A) - V_{a_t^T}(\pi_{\theta^S}^S)$
  Update $s_{t+1}^T \leftarrow s_t^T \cup \{(a_t^T, r_{a_t^T}^S)\}$
  $\pi^S \leftarrow \pi^S + \alpha^S \nabla_{\pi^S} J^S(\pi^S)$
  $\pi^A \leftarrow \pi^A + \alpha^A \nabla_{\pi^A} J^A(\pi^A)$
  $\pi^T \leftarrow \pi^T + \alpha^T \nabla_{\pi^T} J^T(\pi^T)$
  $t \leftarrow t + 1$
end while
return $\pi_{\theta^S}^S$, $\pi_{\theta^A}^A$, and $\pi_{\theta^T}^T$

3) Teacher Agent: The teacher agent in GACL monitors two key elements to design adaptive curricula: student performance and task history. The teacher maintains a history of student rewards, $\{r_i^S\}_{i=0}^{t-1}$, stored in its state s_t^T to monitor student performance. By incorporating these past performance metrics, the teacher can dynamically adjust the curriculum to match the student's evolving skill level. The teacher also retains a history of previously assigned tasks $\{a_i^T\}_{i=0}^{t-1}$ in s_t^T and exploits the latent space $\mathcal{Z}$ learned by the VAE trained on partially known real-world data to produce teacher action based on the latent embedding z_t. Based on both student performance and task history, the teacher generates new tasks using z_t for the student by sampling from this latent space using the VAE decoder $G : \mathcal{Z} \to \mathcal{A}^T$, which maps latent vectors to concrete tasks. The teacher then aims to maximize the expected cumulative regret:

$$J^T(\pi_{\theta^T}^T) = \mathbb{E}_{\tau^T \sim \pi_{\theta^T}^T} \left[\sum_{t=0}^{T} \left(\gamma^T\right)^t \textsc{Regret}_{a_t^T}(\pi_{\theta^S}^S, \pi_{\theta_A}^A) \right],$$

where:

- $\tau^T = (s_0^T, a_0^T, s_1^T, a_1^T, ..., s_t^T)$ is a trajectory in the teacher's MDP,
- $\pi_{\theta^T}^T$ is the teacher policy, parameterized by θ^T,
- $\textsc{Regret}_{a_t^T}(\pi_{\theta^S}^S, \pi_{\theta_A}^A) = V_{a_t^T}(\pi_{\theta_A}^A) - V_{a_t^T}(\pi_{\theta^S}^S)$ is the regret for task a_t^T, generated by the teacher at t, and
- $V_{a_t^T}(\cdot)$ is the value function (expected discounted return) of a policy when executing task a_t^T.

4) Maintaining Domain Relevance via Alternating Sampling: Ensuring that learned policies remain aligned with the target deployment domain is crucial for real-world applicability. To this end, GACL interleaves *reference tasks* from $\mathcal{T}_{\text{real}}$ with *synthetic tasks* sampled by the teacher from its latent space by augmenting teacher action a_t^T:

$$a_t^T = \begin{cases} \text{sample from } \mathcal{T}_{\text{real}}, & \text{with probability } \epsilon, \\ \pi_{\theta^T}^T(s_t^T), & \text{with probability } 1-\epsilon, \end{cases}$$

where $\epsilon \in [0, 1]$ controls the probability of selecting a real-world reference task at each step.

Algorithm 1 summarizes the GACL training loop. We first pretrain the VAE on the available real-world tasks $\mathcal{T}_{\text{real}}$ (line 3). The teacher agent then alternates between sampling reference tasks (ϵ probability) and generating new tasks via $\pi_{\theta^T}^T$ (line 7). Both student and antagonist collect trajectories in the selected task (lines 8–9), after which the teacher computes regret and updates its state (lines 10–11). Finally, all three policies (student, antagonist, and teacher) update their parameters according to their respective objectives (lines 12–14). This process continues until convergence.

IV. Experiments

In this section, we evaluate GACL against three baselines: Base RL (no curriculum), Manual (expert) Curriculum Learning (Manual CL), and CLUTR [5]. We test on two challenging domains: BARN Navigation [6], [9] and Quadruped Locomotion [7]. We also conduct ablation studies by removing each of our three main components to analyze their contributions. Here, we describe our experimental setup, evaluation metrics, results, and curriculum visualizations.

A. Experimental Setup

We conduct experiments in two challenging robotics domains: (1) BARN Navigation in highly constrained environments, and (2) Quadruped Locomotion in confined 3D spaces. Both tasks are implemented in simulation, using partial domain knowledge to ground curriculum generation. Specifically, we use procedurally generated environments to represent the final target deployment domain, but the generation procedure is unknown to the teacher agent, which needs to ground its curriculum through only partial knowledge. Below, we detail each domain's setup, instantiate the student and teacher agent, and summarize the key hyperparameters.

1) BARN Navigation:

a) Environment: We adopt the BARN map generator to produce highly constrained 2D navigation maps, featuring narrow corridors and cluttered layouts [9]. For this experiment, we simulate 128 parallel environments in IsaacGym [15] to accelerate data collection and RL training.

b) Student Agent: The student POMDP's observation space $\mathcal{O}^S$ includes the robot's position and orientation, 270° field-of-view, 720-dimensional LiDAR scans, and the relative goal orientation; action space $\mathcal{A}^S$ comprises continuous linear and angular velocities; and reward function $\mathcal{R}^S$ encourages progress towards the goal while penalizing collisions and excessive time.

c) Teacher Agent: The teacher MDP's action space $\mathcal{A}^T$ is the task latent space, with $s_t^T = \{(a_{t-1}^T, r_{t-1}^S)\}$ tracking the most recent task and performance.

2) Quadruped Locomotion:

TABLE I: Hyperparameters for GACL.

GACL Parameter	**Value**	
Parallel Environments	128	
Latent Task Dimension	32	
Training Epochs	5000	
RL Parameter	**Teacher**	**Student**
Learning Rate	1e-4	3e-4
PPO Epoch	10	5
Discount Factor	0.99	0.99

a) Environment: We simulate a quadruped robot moving in confined 3D environments with rugged terrain and overhanging obstacles and maintain 128 parallel instances.

b) Student Agent: The student POMDP's observation space $\mathcal{O}^S$ encompasses the robot's base pose, joint states (angles and velocities), partial terrain, and ceiling heightmaps; action space $\mathcal{A}^S$ consists of continuous torque commands for each leg; and reward function $\mathcal{R}^S$ incentivizes stable forward motion and penalizes falls, collisions, and undue time expenditure.

c) Teacher Agent: The teacher MDP's action space $\mathcal{A}^T$ encodes both terrain and ceiling configurations in the VAE's latent space, with state representation $s_t^T = \{(a_{t-1}^T, r_{t-1}^S)\}$ following the same structure as navigation.

3) Hyperparameters: Table I summarizes the key hyperparameters used in our experiments.

B. Evaluation Metrics

For evaluation, we employ a comprehensive set of metrics to assess both student performance and curriculum difficulty.

a) Student Performance.: In the BARN Navigation domain, we measure (i) *Success Rate* (the percentage of trials that reach the goal without collisions), (ii) *Navigation Progress* (the proportion of the path traversed before failure), (iii) *Average Steps* per successful episode, (iv) *Average Reward* (summing forward progress and collision penalties), and (v) *Average Speed* (m/s). For Quadruped Locomotion, we track (i) *Success Rate* (no falls before reaching the endpoint), (ii) *Distance Traveled* (m), (iii) *Footstep Efficiency* (ratio of time steps with at least three legs in ground contact), and (iv) *Average Reward* reflecting gait stability and forward progress.

b) Curriculum Difficulty.: To quantify task complexity, we define a domain-specific difficulty score. For BARN Navigation, $D_{\text{nav}} = \alpha\,(\text{path length}) - \beta\,(\text{clearance})$, where path length denotes the shortest collision-free path in the map and clearance is the minimum distance to obstacles along that path. For Quadruped Locomotion, $D_{\text{loc}} = \gamma\,(\text{terrain slope}) + \delta\,(\text{obstacle density})$, where terrain slope measures elevation changes and obstacle density measures how rough and bumpy the terrain is, including the number of uneven features.

V. Results and Discussion

A. Main Results

Table II presents the performance of four methods—Base RL, Manual CL, CLUTR, and GACL—on both the BARN Navigation and Quadruped Locomotion tasks.

TABLE II: Performance Comparison on BARN Navigation and Quadruped Locomotion (mean ± std). ↑ indicates higher is better; ↓ indicates lower is better.

	BARN Navigation				Quadruped Locomotion			
Method	**Success(%) ↑**	**Prog(%) ↑**	**Steps ↓**	**Reward ↑**	**Success(%) ↑**	**Dist.(m) ↑**	**Footstep Eff. ↑**	**Reward ↑**
Base RL	76.16 ± 4.47	64.06 ± 2.38	**36.41** ± 0.15	18.36 ± 0.84	72.34 ± 3.26	4.41 ± 1.57	78.6 ± 2.5	16.74 ± 0.82
Manual CL	76.83 ± 5.02	66.17 ± 2.47	36.65 ± 0.22	19.19 ± 1.17	74.12 ± 4.89	4.67 ± 1.94	79.3 ± 2.7	17.02 ± 1.13
CLUTR	76.67 ± 2.74	66.52 ± 4.23	36.70 ± 0.28	18.39 ± 0.65	74.65 ± 2.31	4.76 ± 1.66	80.2 ± 3.1	17.28 ± 0.95
GACL (Ours)	**81.85** ± 2.51	**68.89** ± 2.77	36.99 ± 0.57	**19.45** ± 0.77	**79.21** ± 2.54	**5.12** ± 1.80	**82.9** ± 2.6	**18.41** ± 0.77

a) BARN Navigation: GACL outperforms all baselines in *Success Rate* (81.85%), *Progress* (68.89%), and *Average Reward* (19.45), demonstrating superior navigation in complex environments. While Base RL achieves the lowest *Average Steps*, it suffers more collisions due to prioritizing speed over safety. Manual CL shows competitive performance but is limited by its fixed progression schedule, and CLUTR's stateless teacher fails to adapt effectively to the student's developing capabilities.

b) Quadruped Locomotion: Similarly, GACL leads in all metrics for the locomotion task: *Success Rate* (79.21%), *Distance Traveled* (5.12 m), *Footstep Efficiency* (82.9%), and *Reward* (18.41). The combination of domain-grounded task tracking and active performance monitoring enables GACL to generate appropriately challenging terrain variations. CLUTR moderately improves over Base RL but is hampered by its lack of domain grounding, while Manual CL provides minor improvements but cannot adapt to the full spectrum of terrain difficulties.

Overall, these results demonstrate GACL's effectiveness in both domains through its adaptive curriculum scheduling based on domain grounding, task tracking, and performance monitoring.

B. Ablation Studies

We conduct ablation studies to evaluate the contribution of each key component in GACL: (i) domain grounding, (ii) task tracking, and (iii) performance monitoring. Each variant removes one of these components from the full GACL framework:

- GACL: The complete GACL framework.
- GACL `w/o grounding`: Removes domain grounding. We discard reference task sampling, relying solely on the teacher agent's synthetic tasks: $a_t^T = \pi_{\theta^T}^T(s_t^T)$.
- GACL `w/o task`: Removes task tracking. We replace the latent task representation a_i^T with a random vector ξ_i in the teacher's state: $s_t^T = \{(\xi_i, r_i^S)\}_{i=0}^{t-1}$, where $\xi_i \sim \mathcal{N}(0, I)$.
- GACL `w/o performance`: Removes performance monitoring. We replace the student's reward r_i^S with a random scalar η_i in the teacher's state: $s_t^T = \{(a_i^T, \eta_i)\}_{i=0}^{t-1}$, where $\eta_i \sim \mathcal{U}(0, 1)$.

Table III shows the *Success Rate* (%) for each ablation variant in both BARN Navigation and Quadruped Locomotion, along with the difference (Diff.) from the full GACL. The full framework consistently outperforms its ablated variants, confirming that each component—domain grounding (`w/o grounding`), task tracking (`w/o task`), and performance monitoring (`w/o performance`)—is essential. Removing domain grounding yields the largest performance drop (-5.49% in Navigation and -5.11% in Locomotion), showing that partial real-world references are key for maintaining task relevance. Omitting performance monitoring or task tracking likewise impairs curriculum adaptation, highlighting the importance of aligning generated tasks with the robot's evolving capabilities.

TABLE III: Ablation Study: Success Rate (%) in Navigation and Locomotion. All differences (Diff.) are relative to the full GACL. Negative values are shown in red.

	Navigation		Locomotion	
Variant	**Success(%)**	**Diff.**	**Success(%)**	**Diff.**
GACL	**81.85**	–	**79.21**	–
GACL `w/o grounding`	76.36	-5.49	74.10	-5.11
GACL `w/o task`	79.86	-1.99	77.31	-1.90
GACL `w/o performance`	77.69	-4.16	75.94	-3.27

C. Curriculum Progression

We visualize how each method evolves task difficulty during training using the heuristic measures described in Section IV-B. Fig. 3 compares Base RL, Manual CL, CLUTR, and GACL in both the BARN Navigation domain (left) and the Quadruped Locomotion domain (right). The *y*-axis represents the difficulty score (higher is more difficult), and the *x*-axis denotes training steps.

For both tasks, Base RL never adjusts task complexity and thus remains effectively uniform throughout training. Manual CL follows a fixed schedule, steadily increasing difficulty from simpler tasks to harder ones. CLUTR hovers near a middle range of task complexity, showing limited capacity to adapt the curriculum to the student's evolving skill. In contrast, GACL dynamically adjusts task difficulty based on student learning stages. Early in training, GACL's difficulty curve starts near a moderate level—reflecting the VAE's initialization around a mean task distribution—then decreases as the teacher learns to manipulate the latent space to produce simpler tasks for the struggling student. Once the student becomes more proficient, the teacher progressively increases task difficulty, prompting further skill development. This adaptive scheduling showcases the synergy among GACL's task-awareness, performance tracking, and partial domain grounding, ultimately yielding more effective curricula than competing methods.

Fig. 4 displays snapshots of the curriculum generated by GACL at four training milestones (25%, 50%, 75%, and 100% of total training). On the left, BARN Navigation tasks

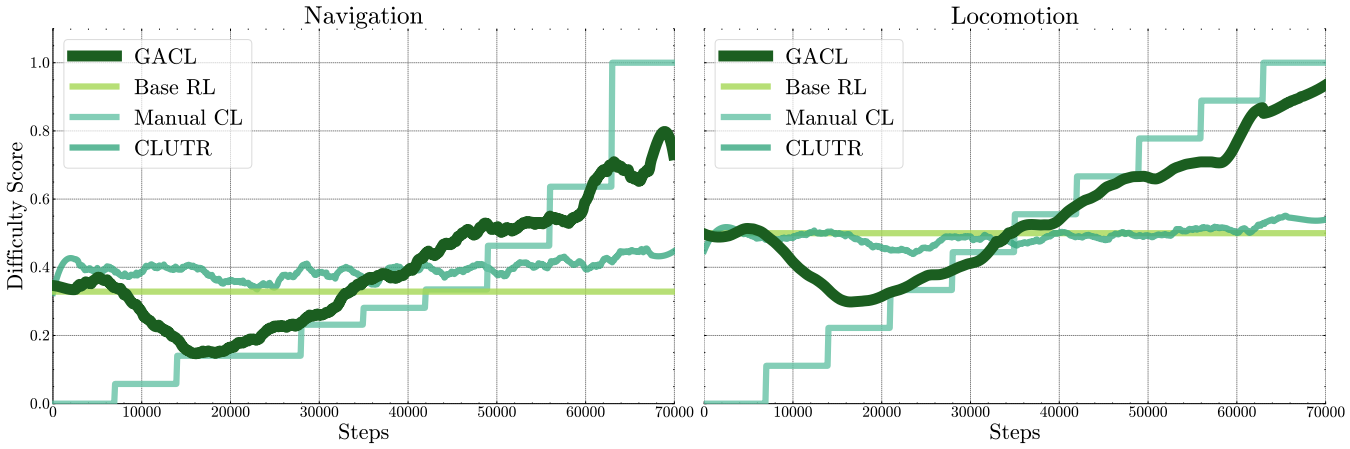

Fig. 3: Curriculum difficulty trends for BARN Navigation (left) and Quadruped Locomotion (right). GACL initially lowers difficulty from a moderate level as it detects the student's struggles, then ramps up complexity once the student becomes more proficient.

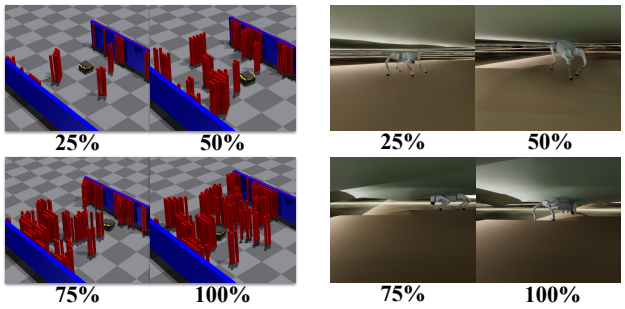

Fig. 4: Progression of tasks generated by GACL over training. Left: BARN Navigation maps sparse to complex. Right: Confined 3D terrains for Quadruped Locomotion, increasing in slope irregularities and obstacles.

evolve from open, sparsely cluttered layouts to narrower corridors populated with numerous obstacles, mirroring the student's increasing competence. On the right, Quadruped Locomotion terrain and ceiling shift from gently sloped surfaces to progressively more uneven and intricate terrain and ceiling obstacles, challenging the robot's growing capabilities. This visual progression illustrates GACL's adaptive approach to curriculum design, where task difficulty increments as the agent's proficiency advances.

D. Discussion

Our experiments demonstrate GACL's effectiveness across two distinct robotics domains with different sensing and control requirements. The ablation studies confirm that all three components—domain grounding, task tracking, and performance monitoring—are essential to GACL's success, as removing any one significantly degrades performance. These results highlight the potential of our unified framework to automatically generate adaptive curricula for complex robot tasks while maintaining focus on relevant target-domain challenges.

VI. Conclusions and Future Work

This paper introduces GACL, a framework specifically tailored to complex robotics domains by actively maintaining domain relevance, task awareness, and performance tracking. Our experiments on two tasks—wheeled robot navigation in the BARN challenge and quadruped locomotion in confined 3D spaces—demonstrate that GACL consistently outperforms both state-of-the-art automated curriculum methods and carefully designed expert curricula. Notably, GACL achieves 6.8% and 6.1% higher success rates in the navigation and locomotion tasks, respectively, compared to SOTA approaches, illustrating its ability to balance structured curricula with flexible adaptation to the robot's evolving capabilities. Through our ablation studies, we observed that each GACL component is critical: *(i)* domain grounding prevents the curriculum from drifting toward unrealistic scenarios, *(ii)* task tracking enables the teacher agent to generate high-dimensional tasks effectively, and *(iii)* performance monitoring ensures the curriculum continually matches the student's learning progress. Together, these elements result in more efficient and robust policy learning for real-world robot deployment.

In future work, GACL can be extended to a broader range of robotic tasks beyond navigation and locomotion, such as manipulation [16], [17] or multi-agent coordination [18]–[21]. Further research could focus on more sophisticated latent-space manipulation and curriculum design, enabling even richer task variations. Another promising direction involves transfer and lifelong learning [22], where pre-trained teacher agents in other domains accelerate adaptation to new ones. Finally, structured learning approaches—from cost-function shaping [23] and kinodynamic modeling [24]–[29] to planner parameter optimization [30]–[35]—could further enhance GACL's efficiency and generalizability in real-world robotic systems.

References

[1] 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.

[2] V. Atanassov, J. Ding, J. Kober, I. Havoutis, and C. Della Santina, "Curriculum-based reinforcement learning for quadrupedal jumping: A reference-free design," *arXiv preprint arXiv:2401.16337*, 2024.

[3] R. Portelas, C. Colas, L. Weng, K. Hofmann, and P.-Y. Oudeyer, "Automatic curriculum learning for deep rl: A short survey," *arXiv preprint arXiv:2003.04664*, 2020.

[4] M. Dennis, N. Jaques, R. Turner, H. Song, J. Z. Leibo, E. Hughes, and M. Botvinick, "Emergent complexity and zero-shot transfer via unsupervised environment design," *arXiv preprint arXiv:2012.02096*, 2020. [Online]. Available: https://arxiv.org/abs/2012.02096

[5] A. S. Azad, I. Gur, J. Emhoff, N. Alexis, A. Faust, P. Abbeel, and I. Stoica, "Clutr: Curriculum learning via unsupervised task representation learning," in *International Conference on Machine Learning*. PMLR, 2023, pp. 1361–1395.

[6] 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.

[7] Z. Xu, A. H. Raj, X. Xiao, and P. Stone, "Dexterous legged locomotion in confined 3d spaces with reinforcement learning," *arXiv preprint arXiv:2403.03848*, 2024.

[8] 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.

[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] 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.

[11] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone, "Curriculum learning for reinforcement learning domains: A framework and survey," *Journal of Machine Learning Research*, vol. 21, no. 181, pp. 1–50, 2020.

[12] R. Campbell and J. Yoon, "Automatic curriculum learning with gradient reward signals," *arXiv preprint arXiv:2312.13565*, 2023.

[13] D. P. Kingma, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[14] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[15] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[16] T. Haarnoja, V. Pong, A. Zhou, M. Dalal, P. Abbeel, and S. Levine, "Composable deep reinforcement learning for robotic manipulation," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 6244–6251.

[17] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke *et al.*, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conference on robot learning*. PMLR, 2018, pp. 651–673.

[18] M. Limbu, Z. Hu, S. Oughourli, X. Wang, X. Xiao, and D. Shishika, "Team coordination on graphs with state-dependent edge costs," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 679–684.

[19] M. Limbu, Z. Hu, X. Wang, D. Shishika, and X. Xiao, "Scaling team coordination on graphs with reinforcement learning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 16 538–16 544.

[20] Y. Zhou, M. Limbu, G. J. Stein, X. Wang, D. Shishika, and X. Xiao, "Team coordination on graphs: Problem, analysis, and algorithms," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024.

[21] B. Liu, X. Xiao, and P. Stone, "Team orienteering coverage planning with uncertain reward," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 9728–9733.

[22] ——, "A lifelong learning approach to mobile robot navigation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1090–1096, 2021.

[23] X. Xiao, T. Zhang, K. M. Choromanski, T.-W. E. Lee, A. Francis, J. Varley, S. Tu, S. Singh, P. Xu, F. Xia, S. M. Persson, L. Takayama, R. Frostig, J. Tan, C. Parada, and V. Sindhwani, "Learning model predictive controllers with real-time attention for real-world navigation," in *Conference on robot learning*. PMLR, 2022.

[24] X. Xiao, J. Biswas, and P. Stone, "Learning inverse kinodynamics for accurate high-speed off-road navigation on unstructured terrain," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 6054–6060, 2021.

[25] H. Karnan, K. S. Sikand, P. Atreya, S. Rabiee, X. Xiao, G. Warnell, P. Stone, and J. Biswas, "Vi-ikd: High-speed accurate off-road navigation using learned visual-inertial inverse kinodynamics," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 3294–3301.

[26] P. Atreya, H. Karnan, K. S. Sikand, X. Xiao, S. Rabiee, and J. Biswas, "High-speed accurate robot control using learned forward kinodynamics and non-linear least squares optimization," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 11 789–11 795.

[27] A. Pokhrel, A. Datar, M. Nazeri, and X. Xiao, "CAHSOR: Competence-aware high-speed off-road ground navigation in SE (3)," *IEEE Robotics and Automation Letters*, 2024.

[28] A. Datar, C. Pan, and X. Xiao, "Learning to model and plan for wheeled mobility on vertically challenging terrain," *IEEE Robotics and Automation Letters*, 2024.

[29] A. Datar, C. Pan, M. Nazeri, A. Pokhrel, and X. Xiao, "Terrain-attentive learning for efficient 6-dof kinodynamic modeling on vertically challenging terrain," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024.

[30] X. Xiao, Z. Wang, Z. Xu, B. Liu, G. Warnell, G. Dhamankar, A. Nair, and P. Stone, "Appl: Adaptive planner parameter learning," *Robotics and Autonomous Systems*, vol. 154, p. 104132, 2022.

[31] X. Xiao, B. Liu, G. Warnell, J. Fink, and P. Stone, "Appld: Adaptive planner parameter learning from demonstration," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4541–4547, 2020.

[32] Z. Wang, X. Xiao, B. Liu, G. Warnell, and P. Stone, "Appli: Adaptive planner parameter learning from interventions," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 6079–6085.

[33] Z. Wang, X. Xiao, G. Warnell, and P. Stone, "Apple: Adaptive planner parameter learning from evaluative feedback," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7744–7749, 2021.

[34] Z. Xu, G. Dhamankar, A. Nair, X. Xiao, G. Warnell, B. Liu, Z. Wang, and P. Stone, "Applr: Adaptive planner parameter learning from reinforcement," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 6086–6092.

[35] D. Das, Y. Lu, E. Plaku, and X. Xiao, "Motion memory: Leveraging past experiences to accelerate future motion planning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16 467–16 474.