arXiv:2608.16164v1 [cs.AI] 17 Aug 2026

Trajectory-Level Automatic Curriculum Learning for Legged Locomotion on Unstructured Terrain

Rocky Liu[1] Tengyu Liu[1] Baoxiong Jia[1] Fangwei Zhong[4] Xinyi Tong[1,2] Hongzhao Xie[3]
Siyuan Huang ✉[1]
[1]National Key Laboratory of General Artificial Intelligence, BIGAI [2]Communication University of China [3]Shanghai Jiao Tong University
[4]Beijing Normal University

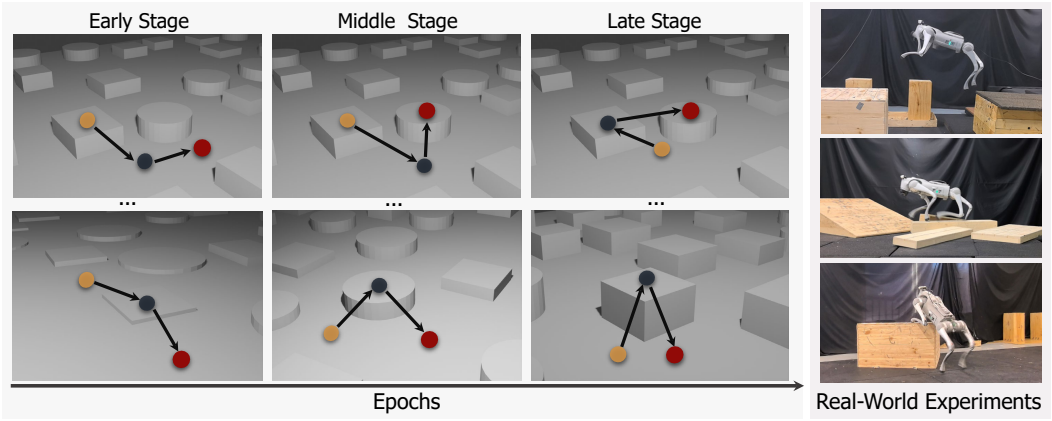


Figure 1: Our method generates effective curricula on unstructured terrain. As training progresses, the sampled tasks evolve with the policy capability, moving toward taller platforms, harder approach angles, descents from elevated structures, and gap-crossing between high platforms. Experiments show that our method improves success rates across terrain tasks, learns more robust behaviors such as platform climbing from diverse approach directions, and supports sim-to-real transfer.

Abstract: Training locomotion policies for complex unstructured terrain requires a curriculum to avoid early exploration failures. However, since unstructured terrain lacks explicit difficulty ordering for curriculum design, existing methods resort to heuristic curricula over parameterized terrains. This abstraction limits generalization, as policies can overadapt to near-fixed perceptual patterns. To address this, we propose **TACL**, an **T**rajectory-level **A**utomatic **C**urriculum **L**earning framework that generates training tasks directly from unstructured terrain maps. At each curriculum update, the evaluator learns a difficulty function for the current policy that maps a given trajectory task to a difficulty score. The sampler then proposes new trajectories guided by the learned evaluator as the curriculum for the next policy update. This forms a closed loop in which the curriculum is iteratively matched to the evolving policy. Quantitative and qualitative experiments show that TACL continuously provides effective curricula on unstructured terrain, improving trajectory success rate by 56.3% over direct training without curriculum. Compared with handcrafted curriculum learning, our method improves success rate by 18.5% on the hardest terrain tasks and by up to 39.74% when evaluating traversal from diverse approach directions on the same obstacle type.

Keywords: Robot Locomotion, Curriculum Learning

1 Introduction

Learning robust locomotion policies for complex terrain with deep reinforcement learning (DRL) [1]. Among the success, curriculum learning is a key role to avoid conservative policies caused by

early-stage exploration failures. However, unstructured maps do not provide an explicit criterion for judging which tasks are suitable for the current policy or how task difficulty should progress. Existing methods address this by imposing structured curriculum spaces.

Handcrafted curriculum learning (HCL) abstracts complex terrain into several parameterized geometric templates [2, 3, 4, 5, 6, 7], e.g., gap terrains, instantiates them along heuristic difficulty progressions, and pairs them with manually specified trajectories for policy training. Despite empirical success in structured settings, HCL requires substantial manual effort and produces limited task distributions, making policies overly adapted to the nearly fixed perceptual patterns along predefined trajectories. Automatic curriculum learning (ACL) reduces the need to manually specify difficulty progressions by adapting task sampling using policy performance or learning signals [8, 9, 10, 11], but it still generate curricula within the same parameterized template spaces rather than directly on unstructured maps. In both HCL and ACL, this abstraction causes terrain information loss and narrows the task distribution, leaving the policy underexposed to the diverse trajectory tasks encountered in unstructured environments. To avoid these changalles that construct curricula directly on unstructured maps, the missing criterion must be learned at the trajectory level. Unlike HCL, which heuristically defines difficulty through predefined terrain-parameter levels, trajectory difficulty on an unstructured map depends on the terrain variations encountered along the path, the transition context, and the current policy capability. The key challenge is therefore to evaluate trajectory difficulty under current policy and sample learnable trajectory-level tasks on a map, without manually designing a terrain parameterization.

In this paper, we propose TACL, an automated curriculum learning framework capable of generating trajectory tracking curricula that match the evolving competence of the policy, enabling effective learning directly on raw unstructured maps. The framework comprises two primary modules: the evaluator and the sampler. At each curriculum update, we roll out the current policy on trajectories randomly sampled from the map, collect success and failure outcomes, and represent each traversed segment by its transition context. We use the conditional variational autoencoder (CVAE) [12] to encode these high-dimensional transition contexts into a compact latent space. The evaluator then learns a policy-conditioned difficulty function that predicts the difficulty of each trajectory from its encoded transition context. Guided by the learned evaluator, the sampler uses Metropolis-Hastings Markov Chain Monte Carlo (MH-MCMC) [13, 14] to search for new waypoint trajectories whose predicted difficulty falls within a suitable range for the current policy. The generated trajectories are used as the curriculum for the next policy update, forming a closed loop in which tasks become progressively harder as the policy improves.

We evaluate our method on both an unstructured terrain map and the handcrafted curriculum map from Extreme Parkour [3], comparing against random trajectory sampling and the HCL baseline. Quantitative and qualitative results show that our method generates effective curricula directly on unstructured terrain, achieving more roboust terrain traversal performance to HCL while improving trajectory-level generalization by 39.5%. Overall, our contributions can be summarized as follows:

- We propose TACL, a closed-loop framework for robot locomotion policy that automatically generates curricula directly on unstructured terrain maps, removing the need for handcrafted terrain curricula and manually parameterized terrain templates.
- We design a context-aware transition encoding and evaluator that accurately predicts task difficulty for current policy, providing reliable guidance for sampling capability-matched training tasks.
- Extensive experiments demonstrate that our framework generates effective curricula directly on unstructured terrain maps, substantially improving policy capability over non-curriculum training and achieving more robust traversal than HCL across diverse approach directions.

2 Related Works

Handcrafted Curriculum Learning in Robot Locomotion. Handcrafted curriculum learning reduces the mismatch between task difficulty and policy capability by manually scheduling training tasks from easy to hard [15]. In standard locomotion, this is often done by expanding command

ranges, such as target velocity, heading, or motion amplitude, enabling skills such as basic locomotion [16, 17, 18], high-speed trotting [19, 20], and jumping [21]. For complex terrain traversal, prior works typically handcraft both terrain curricula and trajectory tasks. They parameterize obstacle families such as stairs, gaps, hurdles, and ramps, and advance the policy through manually specified difficulty sequences using heuristic performance thresholds [16, 2, 3, 4, 22, 23, 5, 6]. Other parkour-oriented methods rely on handcrafted constraints, terrain-specific experts, or predefined waypoint trajectories to learn maneuvers such as crossing gaps, climbing platforms, or traversing ramps [24, 25]. While effective within designed terrain families, these curricula require human expertise to define terrain categories, difficulty progressions, success thresholds, and often the training trajectories themselves. The resulting task distributions can be limited or nearly fixed, making policies overly adapted to specific waypoint paths and perceptual patterns, which reduces transfer to regenerated or less structured terrain instances.

Automatic Curriculum Learning in Robot Locomotion. Automatic curriculum learning aims to reduce manual curriculum design by adaptively selecting training tasks according to the policy's learning state. A central challenge is to identify tasks that are neither too easy nor too difficult for the current policy. Existing approaches use reward prediction, temporal-difference (TD) error, regret, surprise, or learning progress as task-selection signals. HACL learns a reward predictor from historical data for task sampling [26], while unsupervised environment design methods generate increasingly difficult environments through adversarial or evolutionary processes [27, 28, 29, 30]. TD-error- or surprise-based methods prioritize tasks with high prediction error [31, 32, 33], but such signals can mix learnable uncertainty with irreducible noise and trap the policy in unproductive regions [34, 35]. Learning-progress methods instead prioritize task regions where policy performance improves most rapidly [36, 37]. Other studies automate curriculum search within parameterized terrain spaces, where tasks are generated by adjusting terrain sampling probabilities, filtering geometric parameters, discretizing stepping-stone configurations, or updating curriculum ranges in a learned latent space [8, 9, 10, 11]. While these methods automate task selection, they still rely on manually parameterized terrain or task spaces, limiting their applicability to curriculum generation on unstructured maps.

3 Method

As shown in Fig. 2, our framework forms a closed-loop curriculum over trajectory-following tasks. Each curriculum iteration, indexed by k, consists of three coupled processes: (a) policy learning and rollout collection, (b) difficulty evaluator training, and (c) evaluator-guided task generation. Given the current policy π_{θ_k}, rollouts on sampled tasks provide transition-level outcomes for training the evaluator $\mathcal{D}_k$. The updated evaluator guides the sampler $\mathcal{G}_k$ to generate the next task set $\mathcal{S}_{k+1}$, which is used for the subsequent policy update.

3.1 Problem Formulation

We formulate our method as an automatic curriculum learning framework that constructs training task sets for legged locomotion on unstructured terrain. Given a terrain map $\mathcal{M}$, at curriculum iteration k, the framework provides a set of trajectory-following tasks $\mathcal{S}_k = \{\mathcal{T}_j\}$ for policy training. Each task $\mathcal{T} = \{p_0, p_1, \ldots, p_N\}$ specifies one rollout episode, where $p_i \in \mathbb{R}^2$ is a waypoint on $\mathcal{M}$. The robot starts from p_0 and sequentially tracks $p_1, \ldots, p_N$. Consecutive waypoints define sub-trajectories $\tau_i = (p_i, p_{i+1})$, for $i = 0, \ldots, N-1$, where executing τ_i means reaching p_{i+1} from p_i. The policy is trained with DRL: at time t, the actor receives an observation s_t, samples an action $a_t \sim \pi_\theta(\cdot \mid s_t)$, and obtains reward r_t after the action is applied. We use the same waypoint-tracking objective and the same asymmetric actor-critic design as the teacher-stage setup of Extreme Parkour [3], details of observations, rewards, and policy hyperparameters are provided in Appendix.

To evaluate and generate curriculum tasks, we represent each sub-trajectory by a transition context descriptor $\xi_i = (H_p, M_p, H_c, M_c, \delta_i)$. Here (H_p, M_p) denotes the height patch and valid-length

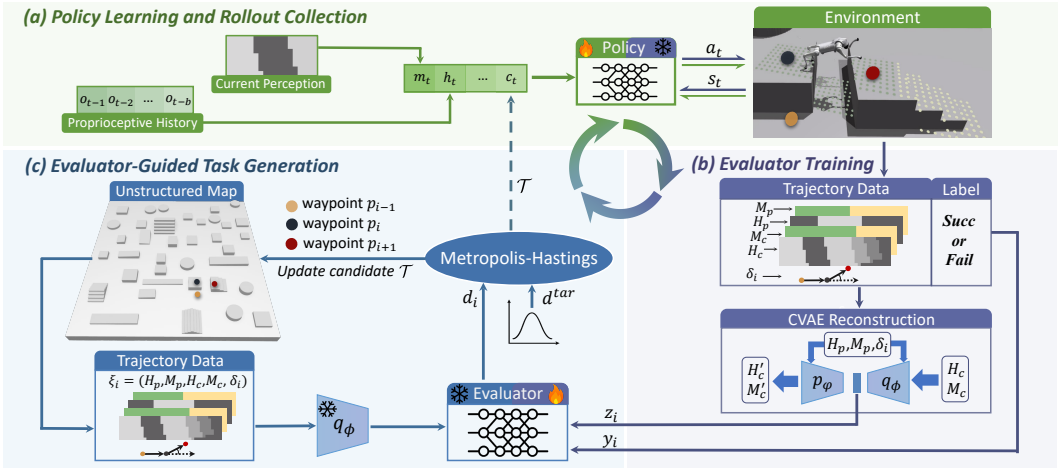


Figure 2: **Framework overview.** (a) The policy learns to follow waypoint trajectories, where a trajectory $\mathcal{T}$ is converted into velocity commands c_t for policy training. (b) The current policy is rolled out on randomly sampled trajectories to collect transition descriptors ξ_i and success/failure labels y_i. The CVAE encoder q_ϕ maps each descriptor to a latent code z_i, and the evaluator predicts its policy-conditioned difficulty d_i. (c) Guided by the updated evaluator, the sampler uses MH-MCMC to generate capability-matched trajectories by sampling for transitions whose predicted difficulty d_i matches the target difficulty d^{tar}. The generated trajectories are used for the next policy update, closing the curriculum loop. Fire and snowflake icons indicate training and inference modes.

mask of the previous sub-trajectory $\tau_{i-1} = (p_{i-1}, p_i)$, and (H_c, M_c) denotes the corresponding pair of the current sub-trajectory $\tau_i = (p_i, p_{i+1})$, and δ_i is the relative yaw angle from τ_{i-1} to τ_i. The height patches satisfy $H_p, H_c \in \mathbb{R}^{L\times W}$, and the masks satisfy $M_p, M_c \in \{0,1\}^{L\times W}$. Each patch is cropped along its waypoint direction and start-referenced by subtracting the mean elevation near the beginning of its centerline, while the mask marks the cells covered by the actual transition length. For $i = 0$, the previous context is set to a flat patch, a zero mask, and zero yaw.

3.2 Difficulty Evaluator Training

A curriculum update is triggered once the policy reaches an average waypoint-completion threshold κ_{wp} on the current tasks, or once the update period K_{upd} is met. We then evaluate the current policy π_{θ_k} on feasible trajectories randomly sampled from $\mathcal{M}$, collecting transition-level rollout outcomes for evaluator training. For each evaluated sub-trajectory $\tau_i = (p_i, p_{i+1})$, we set $\ell_i = \|p_{i+1} - p_i\|_2$ and $T_i = \ell_i/v + \epsilon$, where ϵ is a small time tolerance, and define

$$y_i = \begin{cases} 0, & \text{if } p_{i+1} \text{ is reached within } T_i, \\ 1, & \text{if timeout, falling, or another failure termination occurs.} \end{cases} \tag{1}$$

The label y_i records the execution outcome of τ_i under the current policy, with 0 for successful traversal and 1 for failure. To balance the training data for $\mathcal{D}_k$, we store evaluated transitions $u_i = (\xi_i, y_i)$ in equal-size success and failure queues. Given these labeled transitions, the evaluator $\mathcal{D}_k$ learns to estimate the difficulty of a proposed sub-trajectory for the current policy π_{θ_k}. It takes a latent transition code z_i as input and predicts a score $d_i = \mathcal{D}_k(z_i) \in [0,1]$, where larger values indicate a higher failure likelihood under π_{θ_k}, and therefore a more difficult transition for the current policy. Instead of predicting this score directly from the high-dimensional transition descriptor ξ_i, we first learn a compact latent representation with a CVAE. Let $x_i = (H_c, M_c)$ denote the current sub-trajectory representation and $c_i = (H_p, M_p, \delta_i)$ denote its transition context. The CVAE consists of an encoder $q_\phi(z_i \mid x_i, c_i)$ and a decoder $p_\varphi(x_i \mid z_i, c_i)$, with prior $p(z_i) = \mathcal{N}(0, I)$. We denote the reconstruction produced by the decoder as $x_i' = (H_c', M_c')$. The CVAE is trained by minimizing

$$\mathcal{L}_{\text{cvae}} = \lambda_H \|M_c \odot (H_c - H_c')\|_2^2 + \lambda_M \text{BCE}(M_c, M_c') + \lambda_{\text{KL}} D_{\text{KL}}\left(q_\phi(z_i \mid x_i, c_i) \,\|\, p(z_i)\right), \tag{2}$$

where $\mathrm{BCE}(\cdot,\cdot)$ denotes point-wise binary cross-entropy. The first term reconstructs height only over the valid region, the second term reconstructs the valid-region mask, and the KL term regularizes the latent space. After CVAE training, the encoder provides z_i from (x_i, c_i), and the evaluator $\mathcal{D}_k$ is trained on these latent codes with binary cross-entropy:

$$\mathcal{L}_{\text{diff}} = -\frac{1}{B}\sum_{i=1}^{B}\left[y_i \log d_i + (1-y_i)\log(1-d_i)\right], \qquad d_i = \mathcal{D}_k(z_i). \tag{3}$$

Since $y_i = 1$ corresponds to failed traversal under π_{θ_k}, d_i measures the difficulty of τ_i for the current policy rather than intrinsic terrain difficulty.

3.3 Evaluator-guided Task Generation

After updating the evaluator $\mathcal{D}_k$, the sampler $\mathcal{G}_k$ generates the next training trajectories by searching over waypoint locations on $\mathcal{M}$. We formulate task generation as energy minimization problem. For a candidate trajectory $\mathcal{T} = \{p_0, p_1, \ldots, p_N\}$, the energy is

$$E(\mathcal{T}) = E_{\text{hard}}(\mathcal{T}) + \underbrace{\sum_{i=0}^{N-1}\left|d_i - d^{\text{tar}}\right|}_{E_{\text{soft}}(\mathcal{T})}, \qquad d_i = \mathcal{D}_k(z_i), \quad z_i = \mathrm{Enc}_\phi(x_i, c_i). \tag{4}$$

Here E_{hard} imposes large penalties on infeasible proposals, including trajectories that leave the map boundary, exceed the valid transition length, or place waypoints in invalid or non-traversable regions. E_{soft} is the curriculum objective that encourages all sub-trajectories in $\mathcal{T}$ to match the assigned target difficulty d^{tar}. For each candidate trajectory, we sample d^{tar} from a truncated Gaussian distribution over $[\alpha, \beta]$, centered at moderate difficulty, to emphasize challenging but learnable tasks. In practice, batch sampling draws multiple targets in parallel to improve coverage within this difficulty band. We optimize $E(\mathcal{T})$ with MH-MCMC for J iterations. At iteration j, the sampler proposes a new trajectory $\mathcal{T}' \sim Q(\mathcal{T}' \mid \mathcal{T})$ by perturbing waypoint locations with decaying Gaussian noise, recomputes the affected transition descriptors, and evaluates $E(\mathcal{T}')$. The proposal is accepted with probability

$$A(\mathcal{T} \to \mathcal{T}') = \min\left(1, \frac{\exp\left(-E(\mathcal{T}')/T_j\right) Q(\mathcal{T} \mid \mathcal{T}')}{\exp\left(-E(\mathcal{T})/T_j\right) Q(\mathcal{T}' \mid \mathcal{T})}\right), \tag{5}$$

where $Q(\mathcal{T}' \mid \mathcal{T})$ is the waypoint proposal distribution and T_j is the sampling temperature. Since our waypoint proposal uses symmetric Gaussian perturbations, we have $Q(\mathcal{T}' \mid \mathcal{T}) = Q(\mathcal{T} \mid \mathcal{T}')$, so the proposal terms cancel in the Metropolis-Hastings ratio. In our implementation, both the proposal noise scale and T_j decay over MH-MCMC iterations. Details of E_{hard}, the proposal distribution, and the decay schedules are provided in Appendix. The target interval $[\alpha, \beta]$ remains fixed across curriculum iterations, but it is interpreted through the current evaluator. As $\mathcal{D}_k$ is updated from new rollouts, terrain that was difficult for an earlier policy can receive a lower score for the improved policy. Thus, minimizing E_{soft} with the same d^{tar} drives the sampler toward transitions that remain challenging for the current policy. The final accepted trajectories become the task set for the next policy update, closing the curriculum loop.

4 Experiment

4.1 Experimental Setup

We design our experiments to answer three questions: **Q1:** Can our TACL framework provide an effective curriculum that guides progressive policy learning across a unstructured terrain map? **Q2:** How does TACL compare with handcrafted curriculum learning method? **Q3:** Is the CVAE-based transition encoding necessary for reliable difficulty estimation?

Terrain Maps. We conduct experiments on two terrain maps. *Map_A* is an unstructured training map with circular platforms, rectangular blocks, and obstacle heights increasing from $0.05\,\mathrm{m}$ to

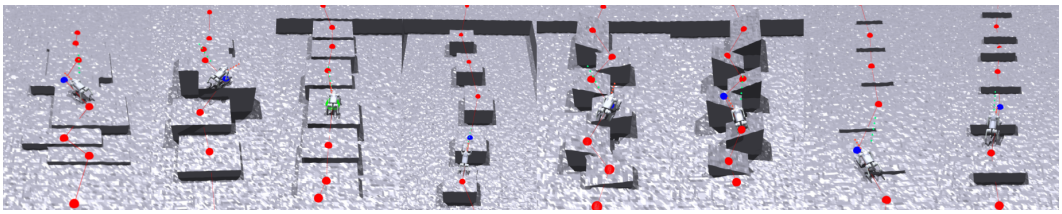
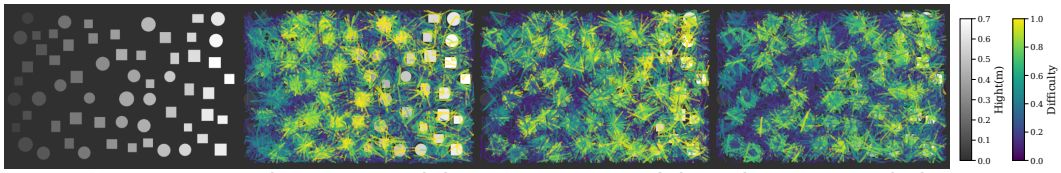


Figure 3: Obstacle families in *Map_HCL*. From left to right: steps, gaps, parkour, and hurdles, each shown at two difficulty levels. Red lines and points denote the predefined waypoint trajectories used in the handcrafted curriculum.



(a) Map_A (b) Trajectories at epoch 2k (c) Trajectories at epoch 4k (d) Trajectories at epoch 16k

Figure 4: Sampler-generated curriculum by our TACL on *Map_A*. (a) The terrain height field. (b)–(d) Trajectories sampled by our method at different policy-training epochs.

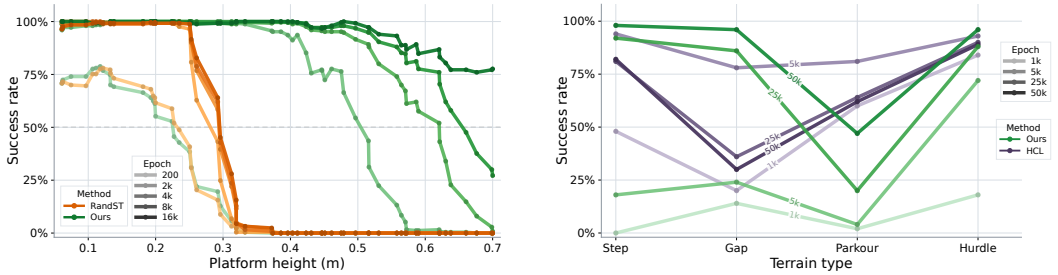
$0.6\,\mathrm{m}$ along the map. *Map_HCL* is a handcrafted curriculum map procedurally instantiated from the Extreme Parkour codebase [3]. It contains flat regions and four obstacle families, including steps, gaps, parkour, and hurdles, as visualized in Fig. 3. Each family is organized into 10 manually specified difficulty levels, with 8 handcrafted sub-terrain instances per level and one predefined 8-waypoint training trajectory per instance.

Baselines. We compare TACL with two baselines. All three methods use the same policy implementation based on the teacher-stage setup of Extreme Parkour [3]. The only difference is the source of training tasks. TACL samples trajectories adaptively according to the evaluated capability of the current policy. RandST trains on feasible trajectories randomly sampled from the given map. HCL-E follows the original handcrafted curriculum setting of Extreme Parkour, where the policy is trained on predefined waypoint trajectories following a manually designed difficulty progression. All policies are trained and evaluated in parallel simulation using Isaac Gym [7]. During evaluation, success is measured at the trajectory level. A rollout is successful only if the robot reaches all waypoints, and it is counted as failed if any sub-trajectory times out, falls, or triggers another failure termination as defined in Eq. 1. Details of all model architectures, hyperparameters, evaluator training, and experimental settings are provided in Appendix.

Real-World Experiments. For real-world deployment, we follow the sim-to-real pipeline of Extreme Parkour [3]. The stage-one policy trained with height-map observations serves as the teacher, and we distill it into a student policy using depth images as exteroceptive input. We deploy the student policy on a Unitree Go2 equipped with an Intel RealSense D435i. This setup isolates our curriculum generation method while using a standard deployment pipeline. Details of stage-two training, deployment, and additional real-world experiments are provided in Appendix.

4.2 Results

A1: TACL generates effective curricula that progressively expands policy capability. We train TACL and RandST on *Map_A* for 16k policy epochs. Fig. 4 visualizes the height map and the trajectories sampled by our method at different epochs. Trajectories are colored by the evaluator-predicted difficulty score, where darker colors indicate easier trajectories and brighter colors indicate harder trajectories. Since the sampler always targets the full difficulty range, the important signal is the spatial location of each difficulty level. At early training, sampled trajectories are limited to flat regions and lower platforms, while the highest platforms are avoided. As the policy improves, the sampled distribution moves toward higher platforms: easy trajectories occupy traversable flat regions, medium trajectories cover platform surfaces, and hard trajectories concentrate near platform



(a) Platform-climbing success rate on *Map_A*.

(b) Evaluation of policies trained on *Map_HCL*.

Figure 5: (a) Our method TACL adapts the training distribution to the evolving policy capability on *Map_A*. (b) Our method TACL achieves comparable results to the handcrafted curriculum baseline on major obstacle families in *Map_HCL*, without using predefined training trajectories.

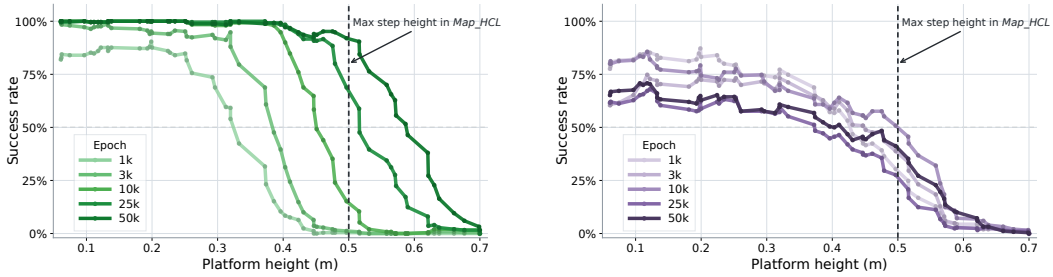
boundaries. This shift shows that our method adapts the task distribution to the current policy capability, producing a learnable curriculum that progressively moves toward harder terrain.

Fig. 5a evaluates policy checkpoints across training epochs on three-waypoint climb-then-descend trajectories, with 50 randomly sampled trials per platform. Our TACL progressively expands the traversable height range, achieving nonzero success across all heights by 4k, over 50% success below 0.65 m by 8k, and above 70% success across all heights by 16k. In contrast, RandST saturates early and converges to a conservative policy whose reliable climbing capability remains around 0.3 m. Across all platform heights, our method improves the average success rate from 39.57% for RandST to 94.54%, a 55.97% gain. These above results show that our method builds a learnable curriculum that progressively improves the policy's traversal capability.

A2: Policies trained with TACL generalize better than HCL across most obstacle families. We train HCL-E, TACL , and RandST on *Map_HCL* for 50k policy epochs. HCL-E follows the handcrafted curriculum protocol of Extreme Parkour [3], using manually specified terrain progressions and predefined 8-waypoint training trajectories. In contrast, TACL treats *Map_HCL* as an ordinary terrain map and adaptively samples training trajectories according to the current policy capability. RandST uses randomly sampled feasible trajectories for training. We evaluate policy checkpoints on separate test terrains generated with the same procedural code and difficulty parameters as *Map_HCL*, but using different random seeds. For each obstacle family, we evaluate the maximum difficulty level on 8 test sub-terrain instances, each paired with a corresponding 8-waypoint trajectory, and report the average trajectory-level success rate over 50 rollouts.

Fig. 5b shows that HCL-E learns quickly from its handcrafted progression but degrades with continued training: from 5k epochs to the final checkpoint, success changes from 94% → 82% on step, 78% → 30% on gap, 81% → 62% on parkour, and 93% → 89% on hurdle. This suggests that HCL-E becomes overly adapted to the predefined trajectories and perceptual patterns. In contrast, our method TACL starts with much lower success at 5k epochs but improves steadily without using handcrafted training trajectories: 18% → 98% on step, 24% → 96% on gap, and 72% → 96% on hurdle. The main limitation is parkour: 4% → 47%, where the trajectory requires a specific human-designed maneuver that is rarely discovered by automatic sampling without additional priors. Overall, across the four maximum-difficulty obstacle families, our method improves the average success rate from 65.75% for HCL-E to 84.25%, an 18.5% gain. RandST achieves 0% success on all obstacle families and is omitted from the plot for clarity.

We further compare TACL with HCL-E on step-task robustness in Fig. 6. In *Map_HCL*, the highest-difficulty step terrain used for training has a height of 0.5 m, which marks the maximum step height observed during training. For each policy checkpoint and platform height, we report the average success rate over trajectories approaching the platform from diverse directions. Within this training height range, TACL improves the overall success rate from 58.68% for HCL-E to 98.42%. Our method shows a capability-frontier expansion strategy, where the policy first improves near its current limit and then shifts the success boundary toward taller platforms. In contrast, HCL-E shows



(a) Ours, requires no predefined training trajectories.

(b) HCL, requires predefined training trajectories.

Figure 6: Platform-climbing evaluation across approach directions. Each value is the average success rate over diverse approach directions, where our method shows more robust traversal than HCL.

limited boundary expansion and even degrades at later epochs, suggesting that predefined trajectories provide insufficient variation in approach directions and traversal contexts.

A3: CVAE-based transition encoding is critical for reliable difficulty estimation and effective task generation. We compare TACL with a variant that removes the CVAE representation. In this variant, the evaluator $\mathcal{D}$ predicts difficulty directly from the raw transition descriptor $\xi_i = (H_p, M_p, H_c, M_c, \delta_i)$: (H_p, M_p) and (H_c, M_c) are concatenated along the channel dimension, encoded by separate convolutional neural networks (CNNs) [38], and fused with δ_i for binary difficulty prediction. The sampler and policy-training pipeline remain unchanged. Table 1 reports evaluator classification accuracy, the average energy of sampled trajectories, and the final trajectory success rate on 1000 randomly sampled feasible trajectories in *Map_A*. Removing the CVAE reduces evaluator accuracy from 92.5% to 67.3%, indicating that direct prediction from raw transition descriptors provides an unreliable difficulty signal. Although the sampler still obtains low energy under this flawed evaluator, the resulting tasks no longer match the capability of the full policy, leading to a success rate drop from 94.6% to 47.3%. This degradation suggests that raw height patches form a non-smooth input space for difficulty prediction, where small geometric perturbations can induce large pixel-level changes and make stable geometry-to-difficulty learning difficult from limited rollout data. The CVAE alleviates this issue by encoding transitions into a more compact and smoother latent space, which is critical for training reliable evaluators and generating tasks that match the curriculum.

Table 1: Ablation of CVAE-based transition encoding on *Map_A*.

Method	Eval. Acc.	Avg. Energy	Success Rate
RandST	–	–	35.2%
TACL w/o CVAE	67.3%	0.013	47.4%
TACL w/ CVAE	**92.5%**	0.021	**94.6%**

5 Conclusion

We propose TACL, an framework for robot locomotion on unstructured terrain, enabling policies to learn complex traversal skills without handcrafted terrain templates or predefined trajectories. The framework forms a closed loop by learning a difficulty evaluator from current-policy rollouts and using it to guide the sampler in generating new training trajectories for subsequent policy learning. Experiments show that our method generates effective curricula on raw terrain maps, improves robot traversal capability and achieves more robust generalization than HCL.

6 Limitations

Although our framework learns robust locomotion behaviors without manually predefined trajectories, highly structured maneuvers may require more directed exploration than generic trajectory sampling provides. Incorporating weak task priors or demonstration-guided proposals could improve sampling efficiency for such behaviors while preserving direct curriculum generation on unstructured maps.

References

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

[2] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. Anymal parkour: Learning agile navigation for quadrupedal robots. *Science Robotics*, 9(88):eadi7566, 2024.

[3] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme parkour with legged robots. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11443–11450. IEEE, 2024.

[4] H. Wang, Z. Wang, J. Ren, Q. Ben, T. Huang, W. Zhang, and J. Pang. Beamdojo: Learning agile humanoid locomotion on sparse footholds. *arXiv preprint arXiv:2502.10363*, 2025.

[5] S. Luo, S. Li, R. Yu, Z. Wang, J. Wu, and Q. Zhu. Pie: Parkour with implicit-explicit learning framework for legged robots. *IEEE Robotics and Automation Letters*, 9(11):9986–9993, 2024.

[6] Y. Dong, J. Ma, L. Zhao, W. Li, and P. Lu. Marg: Mastering risky gap terrains for legged robots with elevation mapping. *IEEE Transactions on Robotics*, 2025.

[7] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State. Isaac gym: High performance gpu-based physics simulation for robot learning, 2021.

[8] Z. Li, C. Li, and M. Hutter. Scaling rough terrain locomotion with automatic curriculum reinforcement learning. *arXiv preprint arXiv:2601.17428*, 2026.

[9] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[10] Z. Xie, H. Y. Ling, N. H. Kim, and M. van de Panne. Allsteps: Curriculum-driven learning of stepping stone skills. In *Proc. ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 2020.

[11] H. Kim, H. Oh, J. Park, Y. Kim, D. Youm, M. Jung, M. Lee, and J. Hwangbo. High-speed control and navigation for quadrupedal robots on complex and discrete terrain. *Science Robotics*, 10(102):eads6192, 2025.

[12] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

[13] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6): 1087–1092, 1953.

[14] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

[15] X. Wang, Y. Chen, and W. Zhu. A survey on curriculum learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):4555–4576, 2021.

[16] B. Tidd, N. Hudson, and A. Cosgun. Guided curriculum learning for walking over complex terrain. *arXiv preprint arXiv:2010.03848*, 2020.

[17] B. Qin, Y. Gao, and Y. Bai. Sim-to-real: Six-legged robot control with deep reinforcement learning and curriculum learning. In *2019 4th International Conference on Robotics and Automation Engineering (ICRAE)*, pages 1–5. IEEE, 2019.

[18] A. Kumar, Z. Fu, D. Pathak, and J. Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

[19] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal. Rapid locomotion via reinforcement learning. *The International Journal of Robotics Research*, 43(4):572–587, 2024.

[20] T. He, C. Zhang, W. Xiao, G. He, C. Liu, and G. Shi. Agile but safe: Learning collision-free high-speed legged locomotion. *arXiv preprint arXiv:2401.17583*, 2024.

[21] V. Atanassov, J. Ding, J. Kober, I. Havoutis, and C. Della Santina. Curriculum-based reinforcement learning for quadrupedal jumping: A reference-free design. *IEEE Robotics & Automation Magazine*, 32(2):35–48, 2024.

[22] J. He, C. Zhang, F. Jenelten, R. Grandia, M. Bächer, and M. Hutter. Attention-based map encoding for learning generalized legged locomotion. *Science Robotics*, 10(105):eadv3604, 2025.

[23] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on robot learning*, pages 91–100. PMLR, 2022.

[24] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot parkour learning. *arXiv preprint arXiv:2309.05665*, 2023.

[25] Z. Zhuang, S. Yao, and H. Zhao. Humanoid parkour learning. *arXiv preprint arXiv:2406.10759*, 2024.

[26] P. Mishra, A. H. Raj, X. Xiao, and D. Manocha. Hacl: History-aware curriculum learning for fast locomotion. *arXiv preprint arXiv:2505.18429*, 2025.

[27] R. Wang, J. Lehman, J. Clune, and K. O. Stanley. Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions. *arXiv preprint arXiv:1901.01753*, 2019.

[28] J. Parker-Holder, M. Jiang, M. Dennis, M. Samvelyan, J. Foerster, E. Grefenstette, and T. Rocktäschel. Evolving curricula with regret-based environment design. In *International Conference on Machine Learning*, pages 17473–17498. PMLR, 2022.

[29] M. Dennis, N. Jaques, E. Vinitsky, A. Bayen, S. Russell, A. Critch, and S. Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. *Advances in neural information processing systems*, 33:13049–13061, 2020.

[30] L. Wang, Z. Xu, P. Stone, and X. Xiao. Gacl: Grounded adaptive curriculum learning with active task and performance monitoring. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 591–596. IEEE, 2025.

[31] M. Jiang, E. Grefenstette, and T. Rocktäschel. Prioritized level replay. In *International Conference on Machine Learning*, pages 4940–4950. PMLR, 2021.

[32] M. Jiang, M. Dennis, J. Parker-Holder, J. Foerster, E. Grefenstette, and T. Rocktäschel. Replay-guided adversarial environment design. *Advances in Neural Information Processing Systems*, 34:1884–1897, 2021.

[33] T. Xu, C. Pan, and X. Xiao. Vertiselector: Automatic curriculum learning for wheeled mobility on vertically challenging terrain. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11136–11143. IEEE, 2025.

[34] 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.

[35] B. Saglam, F. B. Mutlu, D. C. Cicek, and S. S. Kozat. Actor prioritized experience replay. *Journal of Artificial Intelligence Research*, 78:639–672, 2023.

[36] R. Portelas, C. Colas, K. Hofmann, and P.-Y. Oudeyer. Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments. In *Conference on Robot Learning*, pages 835–853. PMLR, 2020.

[37] W. Zhao, Z. Li, and J. Pajarinen. Learning progress driven multi-agent curriculum. *arXiv preprint arXiv:2205.10016*, 2022.

[38] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.