TransCurriculum: Multi-Dimensional Curriculum Learning for Fast & Stable Locomotion

Prakhar Mishra[1,*], Amir Hossain Raj[2], Xuesu Xiao[2], and Dinesh Manocha[1]

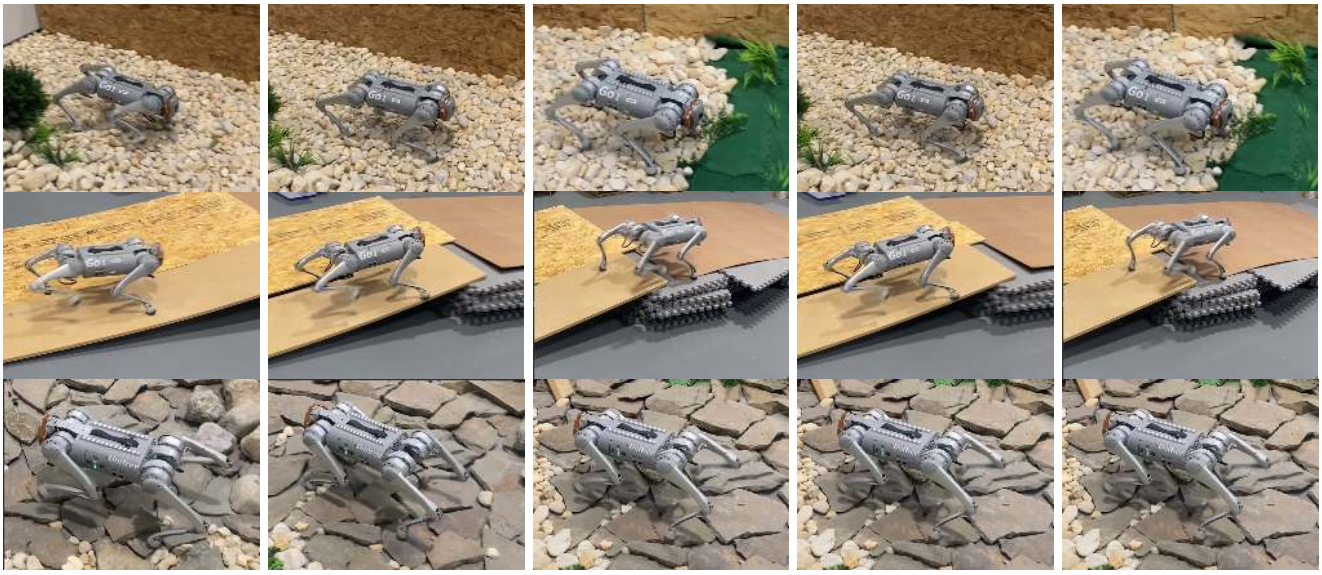

Fig. 1: **Zero-shot hardware evaluation on diverse terrain using Unitree Go1 (*TransCurriculum*).** We report speed, success rate, and lateral deviation over short runs. Row 1: On pebbles (2–3 m), the Go1 maintains 2.1 ± 0.3 m/s with a 60% success rate and 0.3 ± 0.1 m lateral deviation. Row 2: On a wooden slope (approximately $20°$, 3–5 m), it achieves 3.1 ± 0.4 m/s with an 80% success rate and 0.5 ± 0.3 m lateral deviation. Row 3: On rocks (2–3 m), it achieves 1.5 ± 0.4 m/s with a 50% success rate and 0.34 ± 0.1 m lateral deviation. The policy remains functional without fine-tuning, although performance degrades as terrain difficulty increases.

***Abstract*— High-speed legged locomotion struggles with stability and sim-to-real transfer loss at higher command velocities. One key reason is that most curricula vary difficulty along a single axis, for example, by increasing the range of command velocities, terrain difficulty, or domain parameters (e.g., friction or payload mass), using either fixed update rules or instantaneous rewards while ignoring the history of the robot's training progress. We propose *TransCurriculum*, a transformer-based multidimensional curriculum-learning approach for agile quadrupedal locomotion. TransCurriculum adapts along three key axes: velocity-command targets, terrain difficulty, and domain-randomization parameters, including friction and payload mass. Rather than feeding task-reward history directly into the low-level control policy, our formulation uses it at the curriculum level. A transformer-based teacher retrieves a sequence of observed rewards and predicts reward to guide curriculum expansion, while success and learning progress serve as auxiliary prediction objectives. We validate our approach using the Unitree Go1 in Isaac Gym and through zero-shot deployment on hardware. The command-only TransCurriculum configuration achieves a maximum simulated velocity of 6.3 m/s, while the full multidimensional configuration achieves 5.8 m/s with the highest stability score among the evaluated curriculum configurations. We evaluate the TransCurriculum-trained policy on various real-world terrains, including carpet, slopes, tiles, and concrete, achieving a forward velocity of 4.1 m/s on carpet. Table II provides contextual comparisons with zero-shot hardware results reported for prior curriculum methods evaluated on different robotic platforms. The multidimensional curriculum also reduces transfer loss from 27% to 18% relative to the command-only curriculum, demonstrating the benefits of joint training over velocity, terrain, and domain-randomization dimensions while maintaining task success rates of 80–100% on rigid indoor and outdoor surfaces.**

[1]University of Maryland, College Park, MD, USA.
[2]George Mason University, Fairfax, VA, USA.
*Collaborating Researcher
Project page: TransCurriculum.

I. INTRODUCTION

High speed legged locomotion in unstructured environments remains challenging because the performance is subject to command velocity, terrain diversity, and unmodeled dynamic properties such as unknown friction, uneven terrain, slippery or inclined surfaces, etc. In the real world, legged robots operate without complete and accurate knowledge of these real world properties, which causes transfer loss, low task success rate, instability, or outright failure. Model-based controllers can model these environment parameters which can be effective in improving locomotion, but designing them

requires substantial human expertise and may still fail to capture the full contact-rich dynamics [1], [2], [3]. Recent reinforcement learning based methods have been shown to reduce this dependence on human experts in modeling these dynamics and have demonstrated strong robot performance on locomotion tasks such as running or walking, but training high speed RL policies remains a challenge as command velocity and environmental complexity increases [4], [5], [6], [2], [7].

Curriculum learning has emerged as a bridge to improve high speed and stable locomotion task training [8]. It has been used for an array of legged locomotion tasks such as command tracking, terrain difficulty progression, and other tasks such as gait type, climbing, etc [4], [9], [10], [5]. However, one of the key limitations of existing curricula is that they adapt along a single dominant axis either command velocity, terrain, or domain parameters utilizing fixed or threshold-based update rules [4], [11], [12]. These methods work in narrow settings but often lose stability and fail to sustain rapid locomotion when the performance is highly dependent on multiple interacting factors such as commanded forward velocity, ground friction, mass, and terrain rather than being completely isolated in real-world.

Another core difficulty with the multidimensional curriculum space is that scheduler cannot detect trends in policy learning and might overfocus on already mastered task regions [4], [13]. So, this motivates us to use a history-aware curriculum to model temporal evolution of task rewards and adjust the curriculum accordingly rather than injecting it directly into the low level controller.

Main Results: To address this, we introduce ***TransCurriculum***, a ***Trans****former-based* ***Curriculum*** *Learning* approach for fast and stable locomotion. TransCurriculum operates on a joint task space that includes commands, terrain, and domain-randomization parameters. Instead of directly injecting the reward history into the low-level actor-critic policy, we use it at the curriculum level. Specifically, we use a transformer-based teacher that retrieves training history and predicts reward, success, and progress for the curriculum bins. Based on the predicted rewards, we adjust bin sampling to improve learning.

- **Method.** Unlike conventional single-axis curricula, TransCurriculum leverages locally retrieved training history to predict candidate-bin reward, success, and progress, over the velocity, terrain, and domain parameters enabling adaptive multidimensional curriculum expansion.
- **Evaluation.** We extensively evaluated TransCurriculum in simulation on the Unitree Go1 robot in an Isaac Gym simulator and finally validated it on the Go1 hardware in real-world conditions across rigid, deformable, and irregular terrains and moderate slopes (Sections IV–V).
- **Outcome.** The command-only configuration achieves 6.3 ± 0.2 m/s in simulation, while the full configuration achieves 5.8 m/s in simulation and 4.1 ± 0.05 m/s on the Go1 robot. The hardware speed is 18.8% higher than the reported CHRL result of 3.45 m/s [14] and approximately 5.1% higher than the reported RLvRL hardware speed of 3.9 m/s [4]. TransCurriculum also improves stability and reduces transfer loss from 27% to 18% relative to the command-only curriculum. Our results show that history-aware scheduling improves forward velocity, while the multidimensional curriculum improves stability and sim-to-real transfer (Sections IV–V).

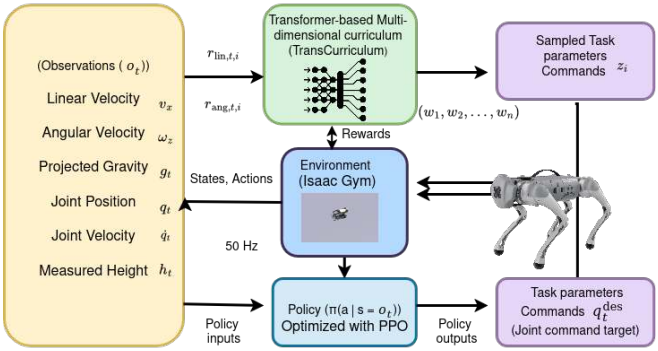

Fig. 2: **TransCurriculum pipeline.** The low-level policy π_θ is trained with **PPO** using rollouts from the Isaac Gym environment. The TransCurriculum module maintains bins over a multidimensional task space comprising commands, terrain difficulty, and domain parameters, and uses rollout observations and rewards to update the sampling distribution. The transformer retrieves context–outcome history and predicts reward to guide curriculum expansion, while success and progress serve as auxiliary objectives. The sampled task context z is applied to the simulator for the next PPO rollout.

II. Related Work

A. Curriculum Learning in Robotics

Curriculum learning has long been used for task stability and sample efficiency by controlling difficulty during training [8]. Previous works include teacher-student curriculum, predefined curriculum, and the automatic curriculum generation for the parametrized environment [15], [16], [17]. Curriculum learning has gained significant traction in robotics research, as these methods provide a useful foundation for robot learning, but they do not address the challenges of fast legged locomotion over command, terrain, and dynamics space [18].

B. Curriculum Learning for legged locomotion

Curriculum learning has gained some traction in legged locomotion; prior work has used it for a wide variety of curriculum strategies, namely command centric curricula for high speed tracking, terrain curriculum for rough terrain adaptation or hindsight [13], [19], [20], [21]. RLvRL [4] uses fixed rules-based schedules, DreamWaQ [22] focuses on terrain aware locomotion, CHRL [14] combines automatic curriculum with hindsight replay to learn locomotion control

policy, but none exploit the reward history like TransCurriculum.

C. Transformer-based sequence modeling and curriculum design

Recent work has shown that transformers [23] can be incredibly helpful for legged locomotion, mainly for policy learning architectures [24]. The Terrain Transformer (TERT) [25] simply uses a transformer at the policy level for simulator-real transfer on diverse terrains. Body Transformer (BoT) [26] introduces embodiment-aware attention for policy learning. At the intersection of curriculum and transformers, CEC [27] injects ordered cross-episode experience and Decision transformer [28] models RL as return conditioned sequence modeling for curriculum like progression. None of these methods models reward training history at multi-dimensional curriculum level for bin selection in locomotion, the core contribution of TransCurriculum.

III. TransCurriculum: Multi-Dimensional Curriculum Learning

Standard curriculum learning for legged robots separates task difficulty along a single axis, such as command velocity [4], [14], terrain difficulty [10], [12], or domain-randomization parameters such as friction and payload mass [14]. Consequently, quadrupedal robots struggle to sustain high, stable velocities as the command-velocity range widens because these dimensions interact in the real world. This decoupled approach may produce higher v_{cmd} velocities under medium or high friction but fail when friction is low. This occurs because the command curriculum does not incorporate friction or payload variation during command-velocity progression. We present *TransCurriculum*, which learns a multidimensional curriculum over the combined task space of commands, domain randomization, and terrain difficulty (1). *TransCurriculum* acts as a curriculum policy (π_{cur}) that selects curriculum bins to guide the low-level control policy (π_θ).

A. Tasks & Design Space

We define the curriculum task space (1) as the combination of command velocities (c), domain parameters (d), and terrain difficulty (t). Here, c represents command-velocity tasks $(v_x^{\text{cmd}}, v_y^{\text{cmd}}, \omega_z^{\text{cmd}})$, d represents domain-randomization parameters such as friction (μ) and payload mass (m), and $t \in [0, 1]$ represents terrain difficulty.

$$z = [c, d, t] \in \mathbb{R}^D \tag{1}$$

This multidimensional curriculum representation allows the policy to reason over multiple sources of difficulty within the task space before proceeding to higher difficulty levels. The curriculum design space is discretized into a grid of bin centroids $G = \{g_i\}_{i=1}^{M}$, where M is the product of the number of bins in each task dimension. In our experiments, we divide the space into $20 \times 10 \times 20 = 4000$ bins; Section IV and Fig. 4 justify this granularity.

B. Curriculum Distribution and Bin Sampling

Each bin i has a weight of w_i representing the current emphasis of the curriculum on that bin, and the sampling probability distribution of the bin i is given by (2):

$$\pi_{\text{cur}}(i) = \frac{w_i}{\sum_{j=1}^{M} w_j} \tag{2}$$

Once a bin i is selected, we draw the specific task context z_i uniformly:

$$i \sim \pi_{\text{cur}}, \qquad z \sim \text{Uniform}(\text{cell}(g_i)) \tag{3}$$

And once the context space z_i is sampled, it is decomposed into the respective command, DR parameters, and terrain difficulty and is instantiated in the simulation environment. As training progresses, the weights w_i of the high performing bins increase based on the predicted rewards of the transformer. In addition, neighboring bins are also assigned higher weights according to their expected rewards, facilitating exploration.

C. Episode Outcomes and EMA based progress tracking

At the end of each episode, we have an outcome vector summarizing policy's performance:

$$y = [r_{\text{lin}},\, r_{\text{ang}},\, f,\, \ell] \tag{4}$$

where r_{lin} & r_{ang} are linear and angular tracking rewards, f is binary fall indicator, and ℓ is the duration of normalized episodes. We derive a binary success indicator label s that demonstrates whether the policy met the minimum tracking thresholds:

$$s = \mathbb{1}[r_{\text{lin}} > \tau_{\text{lin}} \ \wedge \ r_{\text{ang}} > \tau_{\text{ang}}] \tag{5}$$

We also compute a *progress signal* to capture whether π_θ is improving in that region defined as the difference between the current average reward and an exponential moving average (EMA) maintained for each bin, where $\alpha \in [0, 1]$ is the EMA smoothing rate.

$$r = \frac{r_{\text{lin}} + r_{\text{ang}}}{2}, \qquad p = r - \bar{r}_i, \qquad \bar{r}_i \leftarrow \alpha r + (1 - \alpha)\bar{r}_i \tag{6}$$

This progress signal is essential for differentiating local improvement from episode noise, combined with the Transformer's ability to generalize across neighboring bins, enabling the curriculum to focus on most productive bins.

D. History retrieval

We maintain a memory buffer of all past task-outcome pairs stored during training:

$$\mathcal{H} = \{(z_j, y_j)\}_{j=1}^{N} \tag{7}$$

When evaluating the bin with context z, we retrieve its k nearest neighbors from this history.

$$H(z) = \text{KNN}_k(\mathcal{H}, z) \tag{8}$$

And this local retrieval mechanism forces the teacher to learn context-dependent relationships such as how rewards change with terrain or dynamics for the given command values rather than relying on coarse global training values (such as a single EMA over all bins). As a result, this forces the teacher to reason locally and generalize from well-explored regions to their unexplored neighbors.

E. Transformer Teacher

Given a candidate context z and its retrieved local history $H(z)$, the transformer teacher predicts three quantities: expected reward or predicted reward $\hat{r}(z)$, success probability $\hat{s}(z)$, and progress $\hat{p}(z)$:

$$\big(\hat{r}(z),\, \hat{s}(z),\, \hat{p}(z)\big) = f_\psi\big(H(z), z\big) \tag{9}$$

Only the predicted reward $\hat{r}$ directly updates the curriculum weights through (13). The success and progress heads provide auxiliary supervision for learning the history representation, while observed success determines the candidate bins through (10). The history tokens encode the k retrieved context-outcome pairs (z_j,y_j), capturing the temporal evolution of training performance in neighboring bins. The query token attends to the retrieved tokens via cross-attention, enabling the teacher to infer whether the given bin is likely to be useful, already mastered, or still improving.

Why a Transformer? A feedforward network (MLP) receives tracking rewards as input and predicts bin features, but it cannot model how the reward for each bin evolves over time. A recurrent neural network (e.g., an RNN) can model this evolution through a hidden state but may still miss informative past outcomes. A transformer's cross-attention mechanism helps identify patterns across different time scales. Our ablation (Table III) shows that history-aware architectures (Transformer and RNN) substantially outperform the non-history-aware MLP, while the Transformer achieves better speed, stability, and task success rate than the RNN.

F. Reward-Based Curriculum Expansion

TransCurriculum expands the curriculum from bins in which the policy has demonstrated empirical success, grounding the expansion in observed performance rather than solely in transformer predictions. We select bins whose tracking rewards exceed predefined thresholds:

$$S_t = \{i : r_{\text{lin},i} > \tau_{\text{lin}} \ \wedge \ r_{\text{ang},i} > \tau_{\text{ang}}\} \tag{10}$$

We then add the local neighbors $\mathcal{N}(\cdot)$ of the successful bins:

$$C_t = S_t \cup \mathcal{N}(S_t) \tag{11}$$

We use larger neighborhood radii along the velocity-command dimensions to expand more rapidly toward higher command ranges, and smaller radii along the terrain and domain dimensions to increase robustness gradually. For each bin i, the transformer predicts the expected reward:

TABLE I: TransCurriculum Algorithm

Input: curriculum weights w_i, history buffer $\mathcal{H} \leftarrow \emptyset$, transformer f_ψ
for each PPO iteration **do**
  for each parallel environment k **do**
    if episode ended **then**
      Curriculum Update Rule
      1. Successful bins: $\mathcal{S}_t \leftarrow \{r_{\text{lin},i} > \tau_{\text{lin}} \wedge r_{\text{ang},i} > \tau_{\text{ang}}\}$ [Eq. 10]
      2. Build neighbors: $\mathcal{C}_t \leftarrow \mathcal{S}_t \cup \mathcal{N}(\mathcal{S}_t)$ [Eq. 11]
      3. **for** $i \in \mathcal{C}_t$: predict $\hat{r}_i \leftarrow f_\psi(\mathcal{H}(z_i), z_i)$ [Eq. 8, 9, 12]
      4. Update: $w_i \leftarrow \text{clip}(w_i + \beta_0 + \beta_1 \max(\hat{r}_i, 0),\ 0, 1)$ [Eq. 13]
      5. Update EMA, success; append outcomes to $\mathcal{H}$ [Eq. 5–6]
      6. Train f_ψ on current batch [Eq. 14]
      Task Sampling
      7. Draw bin $i \sim \pi_{\text{cur}}$, sample $z \sim \text{Uniform}(\text{cell}(g_i))$ [Eq. 2–3]
      8. Instantiate z: set commands, friction, mass, terrain
    end if
    Execute $\pi_\theta(a_t \mid o_t)$
  end for
  PPO update on π_θ *(no gradients to f_ψ)*
end for

$$\hat{r}_i = \begin{cases} \dfrac{\hat{r}_{\text{lin},i} + \hat{r}_{\text{ang},i}}{2}, & \text{if } |\mathcal{H}| \geq k, \\ 1.0, & \text{otherwise} \end{cases} \tag{12}$$

The fallback value of 1.0 is used early in training when the history buffer is too small for reliable local retrieval. As the history buffer grows, the teacher's predictions increasingly guide curriculum expansion.

$$w_i \leftarrow \text{clip}(w_i + \beta_0 + \beta_1 \max(\hat{r}_i, 0),\, 0,\, 1)\,, \qquad i \in C_t \tag{13}$$

Higher predicted rewards produce larger weight increases for the corresponding bins and their neighbors. This design prevents the curriculum from overfitting to local maxima or ignoring underexplored regions.

G. Teacher optimization

We train the transformer teacher using a multi-task loss that supervises three prediction heads (14).

$$\mathcal{L}_{\text{teacher}} = \underbrace{\|\hat{r} - r\|_2^2}_{\mathcal{L}_{\text{reward}}} + \underbrace{\text{BCEWithLogits}(\hat{s}, s)}_{\mathcal{L}_{\text{success}}} + \lambda \underbrace{\|\hat{p} - p\|_2^2}_{\mathcal{L}_{\text{progress}}} \tag{14}$$

The *reward head* uses MSE to minimize the error between the observed reward r_t and predicted reward $\hat{r}(z)$. The *success head* uses binary cross-entropy to predict whether a bin meets the minimum tracking threshold, while the *progress head* uses weighted MSE to predict whether its performance is stagnant, improving, or declining. The predicted reward updates the curriculum weights, while the success and progress heads provide auxiliary supervision for learning the history representation.

IV. Experiments

A. Simulation Environment

Simulation Details. We implement TransCurriculum on top of open-source repositories [4] and [10], using the Isaac

TABLE II: Summary of representative curriculum methods and their reported zero-shot hardware performance.

Method	Task	Signal	History	Robot†	Speed† (m/s)
RMA [19]	None	None	None	A1	—
RLvRL [4]	Commands	Fixed rule	EMA	Mini Cheetah	3.9
Risky Terrains [11]	Terrain	Success	None	ANYmal	2.5
DreamWaQ [22]	Terrain	Heuristics	None	A1	3.0
CHRL [14]	Cmd. + terrain	Hindsight	Replay	Custom	3.45
LP-ACRL [12]	Cmd. + terrain	Progress	Online estimator	ANYmal	2.5
Aractingi [13]	Reward + terrain	Obs. reward	None	Solo12	1.5
TransCurriculum (Ours)	Cmd. + terrain + DR	Pred. reward	Transformer	Go1	**4.1**

† Robot and speed values are reported from the corresponding cited papers and are provided only as contextual cross-paper references. DR = domain randomization; EMA = exponential moving average; Obs. = observed; Pred. = predicted.

Gym simulator [29]. Our primary platform is Unitree Go1 (12 DoF), modeled from the manufacturer's URDF.

Training budget. All our training experiments use 4000 parallel environments at a simulation frequency of 200 Hz ($\Delta t = 5$ ms), with the policy operating at 50 Hz, on a single NVIDIA RTX 4090 laptop-based GPU.

Command Curriculum. Following [4], we also initialize command velocities from the lower range $[-1.0, 1.0]$ and gradually expand by ± 0.5 as the policy starts to improve. Starting with a wide initial range, such as $[-5, 5]$, destabilizes learning [9], [19], whereas a narrow initial range tends to produce more stable and faster locomotion (Fig. 3).

Discretization. The joint task space is normalized to $[-1, 1]$ and divided into $20 \times 10 \times 20$ bins, generating a total of $M = 4000$ bins. We ablate this choice across 250, 1000, 4000 and 6000 bins and found that 4000 is the sweet spot for sample efficiency and speed optimization (Figure 4).

Domain Randomization (DR). To facilitate sim-to-real transfer, we randomize key dynamics parameters (e.g. friction and payload mass) [30]. Rather than widening these ranges, which can produce conservative policy behavior [4], [10], TransCurriculum includes these DR ranges along the curriculum axis (1), thereby controlling when the policy is exposed to harder dynamics parameters based on demonstrated competence.

B. Experimental Setup (Policy Architecture)

Teacher appears in two senses here: TransCurriculum Teacher selects task difficulty; while privileged teacher policy (π_T) selects actions, without sharing any parameters or gradient.

Teacher policy. Following [4], [10], teacher policy (π_T) receives the current observation o_t and the privilege environment parameters (e.g., friction, restitution, base mass, joint friction, etc.) encoded by a learned embedding e_θ passed through a multilayer perceptron of size $[512, 256, 128]$ to produce joint commands.

Student policy. For deployment, the student policy (π_s) replaces e_θ with an implicit estimate from the history of the last m time steps $h_t = [x_{t-m}, x_{t-1}]$ performing system identification [19], [31] to mimic the teacher (π_T).

Reward Isolation. We do not feed reward history to either π_T or π_S, it is used specifically by TransCurriculum module for bin selection. This ensures that performance gain is attributed to smart task scheduling at the curriculum level rather than additional policy input.

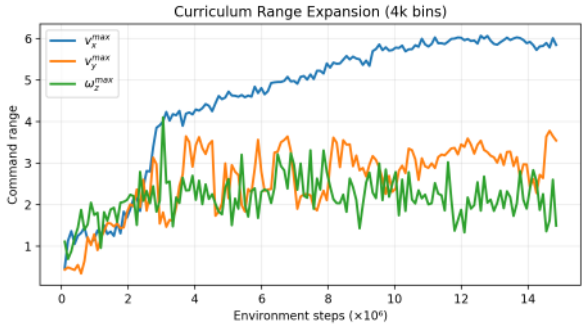

Fig. 3: **Curriculum range expansion over command space:** TransCurriculum expands the initial command range of $[-1.0, 1.0]$ in increments of ± 0.5 based on stable velocity tracking. The maximum sampled commands, $v_x^{\max}$, $v_y^{\max}$, and $\omega_z^{\max}$, show that the forward-speed range expands steadily, whereas the lateral and yaw-rate ranges saturate at lower values, consistent with the forward-locomotion reward design. Actual environment-step counts are ten times the displayed x-axis values.

Optimization. The privileged teacher policy is optimized via PPO [32], while the student policy is trained to mimic it.The TransCurriculum teacher is trained using (14) and acts as a meta-level sampler for curriculum expansion.

C. Evaluation Metric

Cost of Transport (CoT) [33]. The ratio of energy or power consumed by each joint to distance traveled (15):

$$\mathrm{CoT} = \frac{\int_0^T \sum_{j=1}^N |\tau_j(t)\, \dot{q}_j(t)|\, dt}{mg\, \Delta s}. \tag{15}$$

where $\tau_j(t)$ and $\dot{q}_j(t)$ are the torque and angular velocity of joint j at time t, respectively, and $mg\,\Delta s$ is the robot's weight multiplied by the distance traveled. Lower CoT indicates more energy-efficient locomotion.

Stability score (S). It is given by (16):

$$S = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} W_k\, r_k(i), \tag{16}$$

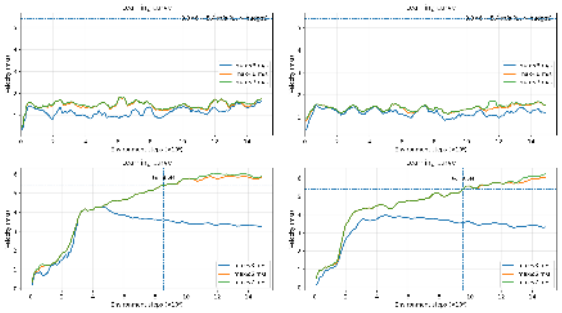

Fig. 4: **Effect of curriculum bin count:** We train command-only TransCurriculum configurations with 250, 1000, 4000, and 6000 bins under identical conditions and compare their learning curves over the first 1500 PPO updates. Actual environment-step counts are ten times the displayed x-axis values. Coarse configurations (250/1000 bins) plateau at approximately 1.5–2.0 m/s. Within this window, the 4000-bin configuration reaches approximately 6 m/s and 90% of the target speed within 86M environment steps, while the 6000-bin configuration reaches 6–6.3 m/s and 90% of the target speed within 95M steps. Under the complete training budget of approximately 4000 PPO updates, the 4000-bin command-only configuration reaches the 6.3 m/s reported in Tables III and IV. We therefore use 4000 bins unless noted otherwise.

Fig. 5: **Disturbance recovery cases on diverse terrains (zero-shot Go1):** These cases demonstrate that the policy experiences bumps, trips, or lateral deviations and returns to its normal gait within a few seconds. The examples show that the TransCurriculum-trained policy remains functional without additional fine-tuning under disturbances and terrain irregularities (see video).

where $r_k(i)$ captures orientation, base height, angular velocity, linear velocity, joint position / velocity limit, self-collision, and torque limit constraints at each time step, with fixed weights W_k. Higher S indicates greater stability.

Task success rate. The percentage of successful runs completed without a crash, tip, fall, or gait failure. We report the mean over 10 trials (real world) and 5 trials (simulation) with 95% confidence interval.

D. Baselines

Curriculum baselines: We compare our TransCurriculum method against the major curriculum strategies used for locomotion: (i) Rapid-Motor Adaptation (RMA) [19]: no explicit curriculum just gradually increases penalties on fixed schedule; (ii) RLvRL [4]: bin selection based on fixed rules for tracking velocity commands; (iii) Solo12 [13] and Risky terrains[11]: terrain difficulty (iv) CHRL [14]: hindsight based curriculum over command and terrain (v) LP-ACRL [12]: terrain curricula with learning progress signal. None of these approaches considers history modeling and Table II summarizes the curriculum signal and history-modeling for each method.

History & Non-history aware schedules. To isolate the contribution of history modeling, we have compared both history (Transformer, RNN) and non history feedforward networks (MLP) as curriculum scheduler under identical training conditions like bin size, learning rate, PPO parameters, and reward functions. (Table III).

V. Results & Discussion

A. Simulation Learning

At $v_x^{\text{cmd}} = 7.0$ m/s, the command-only TransCurriculum configuration reaches 6.3 ± 0.2 m/s. Compared with RLvRL's reported simulation speed of 5.5 m/s [4], this represents an improvement of 0.8 m/s ($\sim 14.55\%$). The full multidimensional configuration reaches 5.8 m/s in simulation and 4.1 ± 0.05 m/s during zero-shot hardware deployment, while providing better stability and sim-to-real transfer, as shown in Table IV.

B. Zero-shot Hardware Transfer (Unitree Go1)

We deploy our TransCurriculum-trained policy zero-shot on a Unitree Go1 EDU robot (12 actuated joints). The policy runs on-board using IMU, joint encoders, and foot force sensors and outputs 12 joint target commands. Given hardware safety constraints, we tested speeds up to 4.1 ± 0.05 m/s with a task success rate of 90%. For context, RLvRL reports a hardware speed of 3.9 m/s on the MIT Mini Cheetah [4], DreamWaQ reports approximately 3.0 m/s on the Unitree A1 [22], and CHRL reports 3.45 m/s on a custom robot [14].

C. Terrain Robustness

To evaluate robustness beyond the training distribution, we test TransCurriculum on Go1 across five terrain categories: rigid indoor, rigid outdoor, deformable, irregular, and moderate slopes.

Rigid-Indoor: Carpet (reference): 4.1 ± 0.05 m/s with a success rate 90%. Tile: 3.1 ± 0.4 m/s, 100% success; occasional foot slips reduce speed by $\approx 24.4\%$ but do not cause any failure.

Rigid-Outdoor: Cement: 3.3 ± 0.3 m/s with a success rate of 80% (19.5% below carpet).

Deformable: Grass: 1.8 ± 0.2 m/s, 70% success (speed 56.1% below the carpet).

Irregular: On pebbles, the robot achieves approximately 2.1 ± 0.3 m/s with a success rate of 60% and a lateral deviation of 0.3 ± 0.1 m, corresponding to a speed reduction of 48.8% relative to carpet. In broken rocks, it achieves approximately 1.5 ± 0.4 m/s with a success rate of 50% and

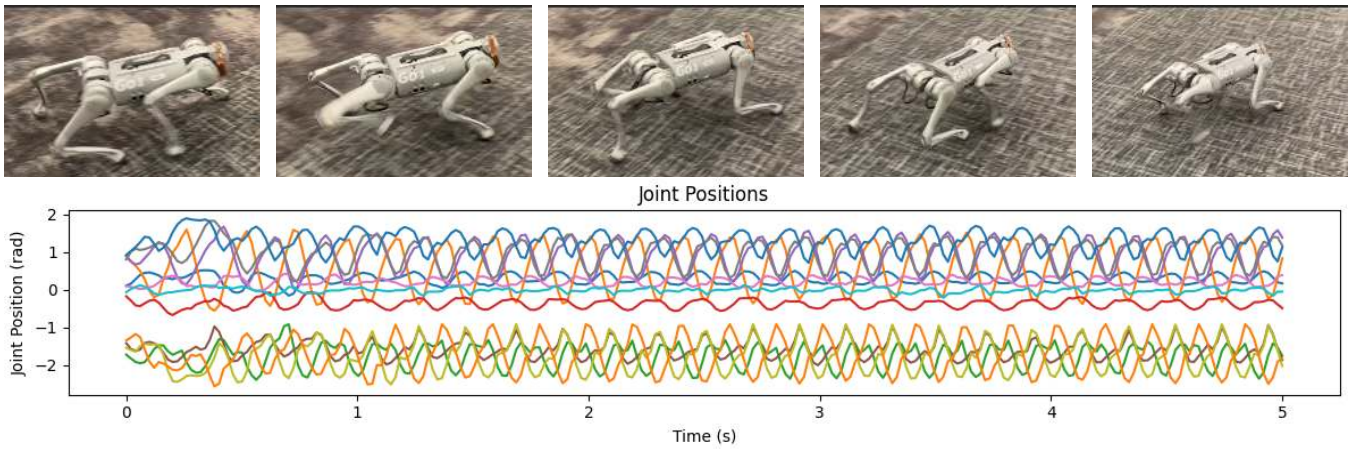

Fig. 6: **High-speed locomotion and stable gait with TransCurriculum. Top:** zero-shot real-world deployment on Unitree Go1 on carpet, policy reaches 4.1 ± 0.05 m/s over a 3-8 m run with 90% task success. **Bottom:** In simulation at $v_x^{\text{cmd}} = 7.0\ \text{m/s}$, the command-only configuration reaches $6.3 \pm 0.2\ \text{m/s}$, and the joint-position trajectories exhibit a consistent periodic pattern, indicating a stable running gait. Together, these results demonstrate that history-aware scheduling enables high-speed training, while multidimensional curriculum sampling improves stability and sim-to-real transfer.

lateral deviation of 0.34 ± 0.1 m, corresponding to a speed reduction of 63.4%.

Slopes: On a cement slope (15°), the robot achieves 2.7 ± 0.3 m/s with a 90% success rate, corresponding to a 34.1% speed reduction relative to carpet. On a wooden slope (20°), it achieves 3.1 ± 0.4 m/s with an 80% success rate, corresponding to a 24.4% speed reduction (Fig. 1).

Performance degrades with terrain compliance and irregularity, resulting in reduced speed, lateral deviation, and occasional failures. However, the system remains functional across all surfaces without any fine-tuning, indicating that multidimensional curriculum produces robust locomotion behavior.

D. Ablation Studies

History ablation (Table III): We compare the history-aware schedulers (Transformer and RNN) with the non-history-aware MLP under identical conditions. At $v_x^{cmd} = 7.0$ m/s, the history-aware methods reach 6.1–6.3 m/s, whereas the MLP reaches only 0.5 m/s and fails the task (0% success). The Transformer outperforms the RNN in speed (6.3 vs. 6.1 m/s), stability (1850 vs. 1800), and task success rate (70% vs. 60%). We attribute the MLP's failure to its inability to model temporal learning dynamics and distinguish useful bins from unproductive ones, leading to poor sampling and progression.

Multidimensionality ablation (Table IV): We isolate the effect of multidimensionality by comparing the command-only, command + DR, and full configurations (command + DR + terrain). Although the command-only curriculum achieves the highest simulated speed (6.3 m/s), the full curriculum improves stability (1850 to 2000), increases the task success rate (70% to 90%), and reduces transfer loss from 27% to 18%, a relative reduction of approximately 33%. These findings suggest that multidimensionality primarily improves stability and transfer, while history-aware scheduling enables high-speed training. Prior work [4], [11], [14] generally treats velocity, terrain, and domain parameters as independent axes; however, these dimensions interact in the real world, and training along a single axis can leave gaps that increase sim-to-real transfer loss.

TABLE III: Non-history vs. History methods ($n = 5$ runs per method)

Metric	Transformer	RNN	MLP
v_x m/s ($v_x^{cmd} = 7$)	6.3 †	6.1	0.5
ω_z rad/s	1.25	1.28	0
History-Aware	Yes	Yes	No
CoT	2.60	5.20	540
Stability (S)	1850	1800	1100
Task Success Rate	70%	60%	0%

† All results use the command-only curriculum configuration.

TABLE IV: Curriculum dimensionality and transfer loss ablation

Metric	Cmd Only	Cmd + DR	Full (ours)
Max sim velocity (m/s)	**6.3**	6	5.8
Stability score (S)	1850	1900	**2000**
Task success rate (%)	70%	70%	**90%**
Sim-to-real † (m/s)	3.65	3.85	**4.1**
Transfer loss	27%	23%	**18%**

† Reported sim-to-real velocities for command velocity of 5 m/s.

E. Failure Modes and discussion

We identify three recurring failure patterns: (i) lateral deviation during straight runs of 2–8 m; (ii) slips and micro-stumbles on low-friction and irregular terrains; and (iii) transient gait irregularities in which the robot switches from trot to crawl for 0.5–1.0 s due to contact-timing errors. These failures likely arise from (i) imperfect reward shaping for heading and straight-line tracking; (ii) substrate mismatch, such as deformable or loose surfaces (e.g., pebbles, tiles, and broken rocks) that were not modeled in the simulator, reducing traction and causing slips; and (iii) contact-inference failures that cause brief gait dropouts. Despite

these failures, the policy recovers quickly from bumps and slips and maintains functional locomotion across all tested terrains. Addressing these failure modes through improved terrain modeling, reward shaping, and latent-aware control is promising future work.

VI. Conclusion, Limitations, and Future Work

We presented **TransCurriculum**, a multidimensional, history-aware, transformer-based curriculum that enables fast and stable quadrupedal locomotion. We demonstrate that curriculum expansion over the joint space of velocity commands, domain parameters, and terrain difficulty improves stability and reduces sim-to-real transfer loss from 27% for the command-only configuration to 18% for the full TransCurriculum configuration. In simulation, the command-only configuration achieves $6.3\ \mathrm{m/s}$, while the full multidimensional configuration achieves $5.8\ \mathrm{m/s}$ and reaches $4.1 \pm 0.05\ \mathrm{m/s}$ during zero-shot transfer to the Go1 robot. Our evaluation focuses on quadrupedal locomotion. We do not study cross-morphology transfer or bipedal/humanoid locomotion [34]. Future work includes morphology generalization, humanoid locomotion, and curriculum strategies for contact uncertainty and terrain-induced failures.

References

[1] M. Zucker, J. A. Bagnell, C. Atkeson, and J. Kuffner, "An optimization approach to rough terrain locomotion," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 3589–3595.

[2] 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.

[3] K. Yin, K. Loken, and M. Van de Panne, "Simbicon: Simple biped locomotion control," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, pp. 105–es, 2007.

[4] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid locomotion via reinforcement learning," *The International Journal of Robotics Research*, vol. 43, no. 4, pp. 572–587, 2024.

[5] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter, "Advanced skills by learning locomotion and local navigation end-to-end," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2497–2503.

[6] Z. Xie, H. Ling, N. Kim, and M. van de Panne, "Allsteps: curriculum-driven learning of stepping stone skills," in *Computer Graphics Forum*, vol. 39, no. 8. Wiley Online Library, 2020, pp. 213–224.

[7] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," *arXiv preprint arXiv:2004.00784*, 2020.

[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] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," in *Conference on Robot Learning*. PMLR, 2023, pp. 22–31.

[10] 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*. PMLR, 2022, pp. 91–100.

[11] C. Zhang, N. Rudin, D. Hoeller, and M. Hutter, "Learning agile locomotion on risky terrains," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 11 864–11 871.

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

[13] M. Aractingi, P. Léziart, T. Flayols, J. Perez, T. Silander, and P. Souères, "Controlling the solo12 quadruped robot with deep reinforcement learning," *scientific Reports*, vol. 13, no. 1, p. 11945, 2023.

[14] S. Li, G. Wang, Y. Pang, P. Bai, S. Hu, Z. Liu, L. Wang, and J. Li, "Learning agility and adaptive legged locomotion via curricular hindsight reinforcement learning," *Scientific Reports*, vol. 14, no. 1, p. 28089, 2024.

[15] T. Matiisen, A. Oliver, T. Cohen, and J. Schulman, "Teacher–student curriculum learning," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3732–3740, 2019.

[16] X. Wang, Y. Chen, and W. Zhu, "A survey on curriculum learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 9, pp. 4555–4576, 2021.

[17] Z. Xu, A. H. Raj, X. Xiao, and P. Stone, "Dexterous legged locomotion in confined 3d spaces with reinforcement learning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 11 474–11 480.

[18] S. Luo, H. Kasaei, and L. Schomaker, "Accelerating reinforcement learning for reaching using continuous curriculum learning," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

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

[20] Y. Chen, H. Bui, and M. Posa, "Reinforcement learning for reduced-order models of legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5801–5807.

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

[22] I. M. A. Nahrendra, B. Yu, and H. Myung, "Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5078–5084.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[24] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Real-world humanoid locomotion with reinforcement learning," *Science Robotics*, vol. 9, no. 89, p. eadi9579, 2024.

[25] H. Lai, W. Zhang, X. He, C. Yu, Z. Tian, Y. Yu, and J. Wang, "Sim-to-real transfer for quadrupedal locomotion via terrain transformer," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5141–5147.

[26] C. Sferrazza, D.-M. Huang, F. Liu, J. Lee, and P. Abbeel, "Body transformer: Leveraging robot embodiment for policy learning," *arXiv preprint arXiv:2408.06316*, 2024.

[27] L. X. Shi, Y. Jiang, J. Grigsby, L. Fan, and Y. Zhu, "Cross-episodic curriculum for transformer agents," *Advances in Neural Information Processing Systems*, vol. 36, pp. 13–34, 2023.

[28] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," *Advances in neural information processing systems*, vol. 34, pp. 15 084–15 097, 2021.

[29] 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.

[30] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

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

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

[33] A. Schperberg, M. Menner, and S. Di Cairano, "Energy-efficient motion planner for legged robots," *arXiv preprint arXiv:2503.06050*, 2025.

[34] W. Huang, I. Mordatch, and D. Pathak, "One policy to control them all: Shared modular policies for agent-agnostic control," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4455–4464.