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

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

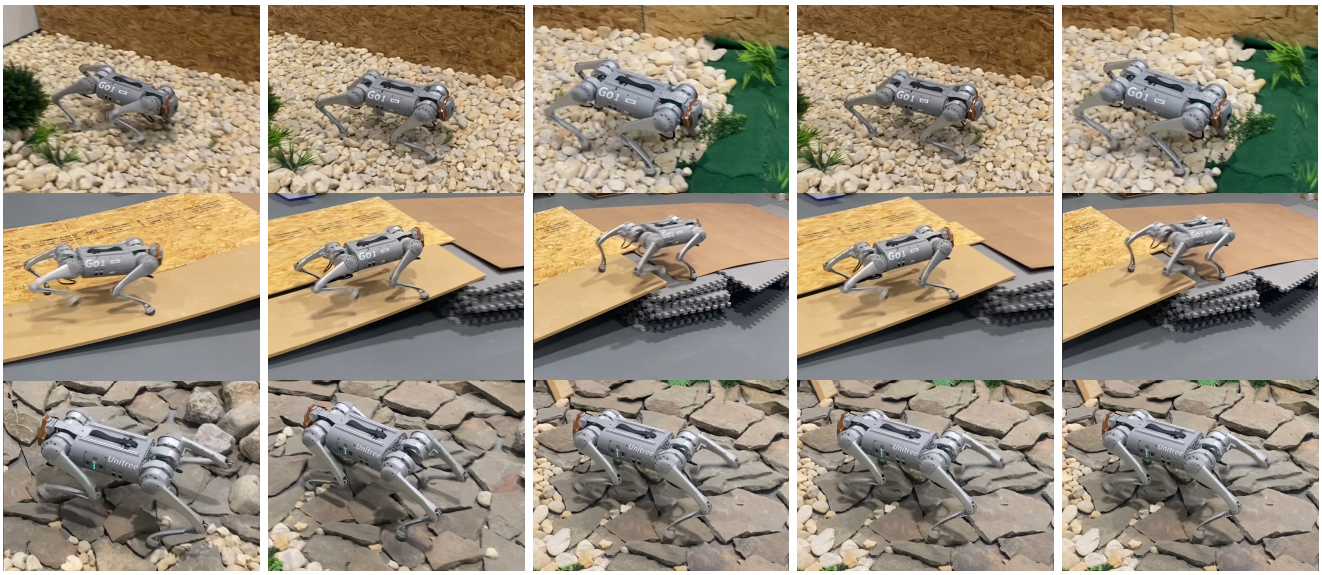

Fig. 1: **Zero-shot hardware evaluation on diverse terrain (Unitree Go1, *TransCurriculum*).** We deploy Transcurriculum on Go1 and report speed, success rate and lateral deviation over short runs. **Row 1:** Pebbles (2-3 m) Go1 maintains 2.1 ± 0.3 m/s, 60% success, 0.3 ± 0.1 m lateral deviation. **Row 2:** Wooden slopes (approximate 20° and 3-5 m) 3.1 ± 0.4 m/s, 80% success, and lateral deviation of 0.5 ± 0.3 m. **Row 3:** Rocks (2-3 m) 1.5 ± 0.4 m/s, 50%, success and 0.34 ± 0.1 m lateral deviation. The policy remains functional across terrain without finetuning, with some performance degradation with terrain difficulty,supporting benefits of history-aware & multi-dimensional training.

***Abstract*— High-speed legged locomotion struggles with stability and transfer losses at higher command velocities during deployment. One of the key reasons is that most curricula vary difficulty along single axis, for example increase the range of command velocities, terrain difficulty, or domain parameters (e.g. friction or payload mass) using either fixed update rule or instantaneous rewards while ignoring how the history of how the robot training has evolved. We propose *TransCurriculum*, a transformer based multi-dimensional curriculum learning approach for agile quadrupedal locomotion. TransCurriculum adapts to 3 key axes, namely velocity command targets, terrain difficulty, and domain randomization parameters (friction and payload mass). Rather than feeding task reward history information directly into the low-level control policy, our formulation exploits it at the curriculum level. A transformer-based teacher retrieves the sequence of observed rewards and uses it to predict the future rewards, success rate, and the learning progress to guide expansion of this multidimensional curriculum towards high performing task bins. And finally we validate our approach on the Unitree Go1 robot in a simulation (Isaac Gym) and deploy it zero-shot on the Go1 hardware. Our TransCurriculum policy achieves a maximum velocity of 6.3 m/s in the simulator and the highest stability score and outperforms prior curriculum baselines. We tested our TranCurriculum trained policy on various real world terrains (carpets, slopes, tiles, concrete), achieving a forward velocity of 4.1 m/s on carpet surpassing the fastest curriculum methods by nearly 18.8% and achieves maximum zero-shot value among all tested methods (Table II). Our multi-dimensional curriculum also reduces the transfer loss to 18% from 27% for command only curriculum, demonstrating the benefits of joint training over velocity, terrain and domain randomization dimension while keeping the task success rate of 80–90% on rigid indoor and outdoor surfaces.**

[1]University of Maryland, College Park, MD, USA.
[2]George Mason University, Fairfax, VA, USA.
*Collaborating Researcher; M.Eng. University of Maryland, 2023.

I. INTRODUCTION

High speed legged locomotion in unstructured environments remains challenging because the performance is subjected 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 expert in modeling these dynamics and have been shown to show 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 completely isolated in real-world.

Another core difficulty with the multi-dimensional curriculum space is that scheduler cannot detect trends in policy learning and might over focus 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 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 which retrieves training history, and predicts reward, success, and progress of the curriculum bins. And based on these predicted rewards, we influence sampling of the bins which improve learning. In this way, TransCurriculum reduces manual schedule design while improving the balance between agility, and stability via utilizing history.

- **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.
- **Approach.** Compared to fixed-rule or threshold based or single-axis curriculum [4], [12], [13], in TransCurriculum, we shift temporal modeling from the low-level control policy to the curriculum level and incorporate a multi-axis approach. And, this enables efficient bin sampling, higher speed, and lower sim-to-real transfer loss (ref. V. Results)
- **Evaluation.** We extensively evaluated TransCurriculum in simulation on the Unitree Go1 robot in an IsaacGym simulator and finally validated it on the Go1 hardware in real-world conditions across various terrains like rigid, deformable, irregular and moderate slopes.(ref. section IV-V)
- **Outcome.** In simulation, TransCurriculum achieves a velocity of 6.3 ± 0.2 m /s and 4.1 ± 0.05 m / s on Go1 robot. This is higher than the CHRL baseline [14] of the command & terrain curriculum (3.45 m / s) by 18.8% and also the non-Go1 curriculum RLvRL (3.9 m / s) [4] with an improvement of 5.15%. TransCurriculum also improves stability and reduces transfer loss from 27% to 18% relative to the command only curriuclum. Our results show that history-aware scheduling helps in improving the forward velocity while the multi-dimensional curriculum approach helps in stability and sim-to-real transfer loss (Section IV-V).

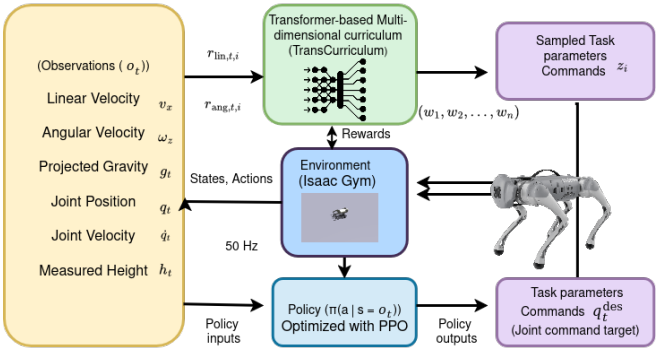

Fig. 2: **TransCurriculum pipeline.** The low-level policy π_θ is trained with **PPO** using rollouts from IsaacGym environment. While the TransCurriculum module maintains bins over multi-dimensional task space of commands, terrain difficulty and domain parameters and use rollouts (observations and rewards) to update distribution. The transformer-curriculum retrieves context-outcome history, to predict the rewards, success and progress of these curriculum bins. These 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]. 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 [17].

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], [18], [19], [20]. RLvRL [4] uses fixed rules-based schedules, DreamWaq [21] 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. Transfomer-based sequence modeling and curriculum design

Recent work has shown that transformers [22] can be incredibly helpful for legged locomotion, mainly for policy learning architectures [23]. The Terrain Transformer (TERT) [24] simply uses a transformer at the policy level for simulator-real transfer on diverse terrains. Body Transformer (BoT) [25] introduces embodiment-aware attention for policy learning. At the intersection of curriculum and transformers, CEC [26] injects ordered cross-episode experience and Decision transformer [27] models RL as return conditioned sequence modeling for curriculum like progression. None of these method model 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 in legged robots separates tasks difficulty along single axis like command curriculum [4], [14], terrain difficulty level [10], [12] or domain randomization [14] parameters like friction, payload mass. Thereby quadrupedal robots struggle to sustain higher and stable velocities as the command-velocity range widens because in the real world these domains interact with each other. This decoupled approach gives a robot higher v_{cmd} velocities with higher or medium friction, but fails when the friction is too low. Because the command curriculum never incorporated friction or payload observation along with progression in command velocities. We present *TransCurriculum* which learns the multi-dimensional curriculum over the combined task space of commands, domain randomization, and terrain difficulty (1). *TransCurriculum* acts as a curriculum-policy (π_{cur}), which selects curriculum bins to guide the low-level control policy (π_θ).

A. Tasks & Design Space

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

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

This multidimensional representation of the curriculum allows the policy to reason over multiple sources of difficulties within the defined task space before proceeding to the higher task difficulty level. And the curriculum design space is discretized into a grid of bins centroids $G = \{g_i\}_{i=1}^{M}$, where M is the product of bin count per task dimension. In our experiments, we divided 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 weight of w_i demonstrating 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.

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

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 $\hat{p}(z)$:

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

The history tokens encode the k retrieved context-outcome pairs (z_j,y_j), capturing the temporal evolution of the training performance of the 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 Transformer? A feedforward network (MLP) received all the tracking rewards input and predict bin features and cannot model how the rewards per bin evolve over time. A recurrent neural network (for e.g. RNN) can model this via hidden layer but might still miss the most effective past outcomes. A transformer's cross attention helps in learning those hidden pattern across different time frame. Our ablation (Table III) also confirms that history-aware architectures (Transformer, RNN) dramatically outperform non-history methods like MLP, while Transformer outperforms in terms of speed, stability and task success rate.

F. Reward-based curriculum expansion

TransCurriculum expands the active curriculum outward from bins where the policy has already demonstrated empirical success. This ensures that our expansion strategy is grounded in the policy rather than solely relying on transformer predictions. We select the bins whose tracking rewards exceed a defined threshold:

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

Next, we add the local neighboring bins ($\mathcal{N}(.)$) of the successful bins:

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

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: $\mathcal{S}_t \leftarrow \{r_{\mathrm{lin},i} > \tau_{\mathrm{lin}} \wedge r_{\mathrm{ang},i} > \tau_{\mathrm{ang}}\}$ [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: $w_i \leftarrow \mathrm{clip}(w_i + \beta_0 + \beta_1 \max(\hat{r}_i, 0),\ 0, 1)$ [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 $i \sim \pi_{\mathrm{cur}}$, sample $z \sim \mathrm{Uniform}(\mathrm{cell}(g_i))$ [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

For curriculum expansion, we used larger neighborhood radii along velocity command dimensions for faster expansion toward higher command range, compared to smaller range along terrain and domain parameters for a gradual robustness growth. For each bin i, our transformer predicts the expected reward:

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

The fallback value of 1.0 helps the curriculum recover in an early stage when the history buffer is too small to make any reliable local retrieval. But as training progress and the history buffer grows, the teacher's prediction increasingly guide the curriculum bin expansion.

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

The predicted rewards of the transformer $\hat{r}_i$ decide the direction of curriculum expansion (higher predicted rewards lead to a higher weight increase for those bins, along with their neighboring bins). This design prevents curriculum from overfitting to local maxima or ignore underexplored regions.

G. Teacher optimization

We train our transformer teacher using a multi-task loss function that supervises 3 prediction heads (14)

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

The *reward head* (MSE Loss) aims to minimize the error between the observed rewards (r_t) and the predicted rewards $\hat{r}(z)$. The *success head* (binary cross-entropy) tracks whether the bins meet the minimum tracking threshold or not. The *progress head* (weighted MSE) predicts whether

TABLE II: TransCurriculum vs. representative curriculum baselines. Zero-shot transfer robot and maximum zero-shot real-world speed are reported when available ([†]).

Method	Curriculum target	Curriculum signal	Curriculum History modeling	Zero-shot[†] robot	Max zero-shot[†] speed (m/s)
RMA [18]	None	None	None	A1	—
RLvRL [4]	Velocity commands	Fixed-rule	EMA	Mini Cheetah	3.9
Risky Terrains [11]	Terrain difficulty	Task success	None	ANYmal	2.5
DreamWaQ [21]	Terrain difficulty	Curriculum heuristics	None	A1	3.0
CHRL [14]	Commands + Terrain	Hindsight success	Replay	Custom	3.45
LP-ACRL [12]	Commands + Terrain	Learning progress	Online estimator	ANYmal	2.5
Aractingi [13]	Reward + Terrain	Obs–Reward	None	Solo12	1.5
TransCurriculum (Ours)	Commands + Terrain + DR	Pred–Reward	Transformer	Go1	**4.1**

[†] Reported zero-shot transfer robot and maximum zero-shot real-world speed from the corresponding cited paper, when available.
Legend: Obs–Reward = observed reward ; Pred–Reward = predicted reward from the curriculum model.

the performance of a particular bin is stagnant, improving, or declining. As the training history grows, the teacher's prediction becomes increasingly accurate and the curriculum expansion becomes more targeted, since we use those signals to update the curriculum weight (w_i). This allows for more efficient bin selection and faster convergence to higher and more stable forward velocities.

IV. Experiments

A. Simulation Environment

Simulation Details. We implement TransCurriculum on top of open-source repositories [4] and [10], using the IsaacGym simulator [28]. 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 control frequency of 200 Hz ($\mathbf{\Delta t} = 5$ ms) on a single Nvidia RTX 4090 laptop-based GPU. Training runs 400 million steps (roughly 4000 PPO updates) in about 4 hours.

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], [18], where as narrow range earlier tend to produce a more stable and faster speed (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)[29]. Rather than widening these ranges, which can produce conservative policy behavior [4], [10], TransCurriculum included 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

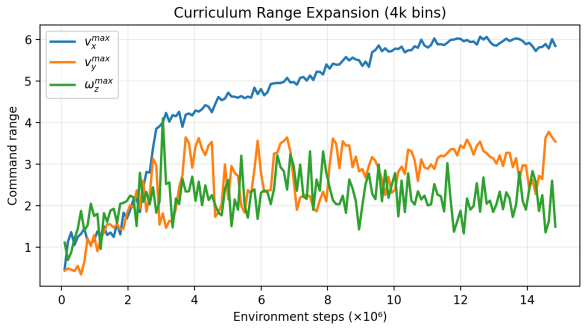

Fig. 3: **Curriculum range expansion over command space:** TransCurriculum starts from a narrow command range of $[-1.0, 1.0]$ and gradually expands by $[-0.5, 0.5]$ as the policy achieves stable velocity tracking performance. We plot the maximum sampled command values during training, $v_x^{\max}$, $v_y^{\max}$ and $\omega_z^{\max}$, as TransCurriculum expands the command range. The curriculum steadily expands along forward speed $v_x^{\max}$, while lateral and yaw-rate saturate at lower thresholds, consistent with our reward shaping for forward locomotion.

(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 (for 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 step $h_t = [x_{t-m}, x_{t-1}]$ performing system identification [18], [30] to mimic the teacher (π_T).

Reward Isolation. We did not feed the 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.

Optimization. Both teacher-student policies are optimized via PPO [31]. TransCurriculum acts as meta-level sampler for curriculum for curriculum expansion.

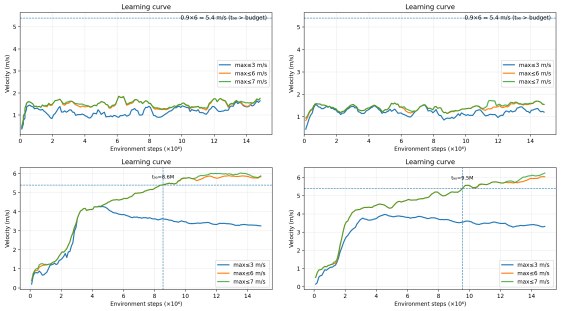

Fig. 4: **Effect of curriculum bin selection criteria:** We train TransCurriculum for 250, 1000, 4000 and 6000 bins under identical training conditions and compare the learning curve (1500 PPO updates). Coarse binning (250/1000) do not reach the high-speed within the given training time, plateauing around 1.5 – 2.0 m/s. The 4000-bin configuration achieves approximately 6 m/s and reaches 90% of target speed within 8.6M environment steps, providing the best tradeoff between exploration and stable curriculum updates. With increasing resolution to 6000 bins increases velocity to $6-6.3$ m/s and the 90% of target speed within 9.5M steps. We therefore use 4000 bins in our experiment unless noted otherwise.

Fig. 5: **Disturbance recovery cases on diverse terrains (zero-shot Go1):** The above cases demonstrates that our policy experiences a bump/trip or deviate laterally, and re-stabilizes to its normal gait within few seconds. Above examples demonstrate that TransCurriculum-trained policy remains functional without any additional finetuning under disturbances or terrain irregularities (See accompanying video).

C. Evaluation Metric

Cost of Transport (CoT) [32]. 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 $\tau_j(i)$ and $\dot{q}_j(i)$ are the torque and angular velocity of the joint j at the time step i, and $mg\,\Delta s$ is the weight times distance traveled. Lower CoT indicates efficient locomotion.

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

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

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) [18]: 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], & 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, & 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. (Tables III).

V. Results & Discussion

A. Simulation Learning

At $v_x^{cmd} = 7.0$ m/s, TransCurriculum reaches 6.3 ± 0.2 m / s and achieves the highest among curriculum baselines in our comparison table II. Compared to other command centric curriculum methods, like RLvRL (5.5 m / s) [4], this corresponds to $+0.8$ m/s ($\sim 14.55\%$) improvement in maximum simulation speed.

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 up to 4.1 ± 0.05 m / s with a task success rate of 90%. This exceeds the previous fastest reported RL-based zero-shot transfer of 3.9 m / s on the MIT mini cheetah [4]. While other methods like DreamWaq[21] report nearly 3.0 m/s on Unitree A1, while CHRL [14] report 3.45 on a custom robot, Transcurriculum outperforms these and other curriculum baselines. Our key curriculum relevant findings and comparison are detailed in Table II.

C. Terrain Robustness

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

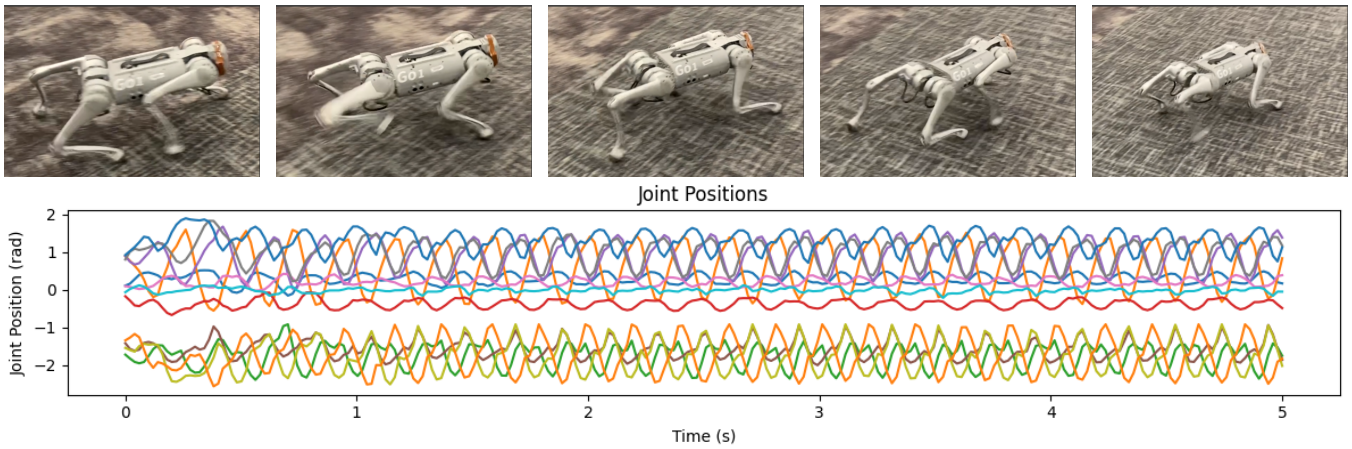

Fig. 6: **High-speed locomotion and stable gait with TransCurriculum. Top:** zero-shot real-world deployment on Unitree Go1 on carpet, policy reaches 4.1 ± 0.05 m/s over a 3-8 m run with 90% task success. **Bottom:** In simulation at $v_x^{cmd} = 7.0$, Go1 reaches 6.3 ± 0.2 m/s and the 12 joint-position time plot shows a consistent periodic pattern, indicating stable running gait at higher command velocity. Together these results substantiate that this is enabled by history-aware and multi-dimensional curriculum task sampling over commands, terrain and dynamics.

Rigid-Indoor: Carpet (reference): 4.1 ± 0.2 m / s with a success rate 90%. Tile: 3.1 ± 0.4 m/s, 100% success; occasional foot slips reduce speed by $\approx 24.0\%$ but does not cause any failure.

Rigid-Outdoor: Cement: 3.3 ± 0.3 m / s with a success rate of 80% (20.0% below carpet).

Deformable: Grass: 1.8 ± 0.2 m / s, 70% success (speed 56.0% below the carpet).

Irregular: On pebbles: $\approx 2.1 \pm 0.3$ m/s and 60% success, with modest drift 0.3 ± 0.1. Broken rocks: $\approx 1.5 \pm 0.4$ m / s and drift 0.3 ± 0.2, 50% success (speed 48. 8% and 63. 4% below carpet).

Slopes: Cement (15°): 2.7 ± 0.3 (90%); Wood (20°): 3.1 ± 0.4 m / s (80%) (see Figures 1) (speed 34. 1% and 24. 0% below carpet).

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 curriculum (Transformer, RNN) against the non-history curriculum (MLP) under identical conditions (Table III). The stark difference is for $v_x^{cmd} = 7.0$ m/s, where a history-aware curriculum reaches 6.1 – 6.3 m/s compared to 0.5 m/s for non-history ones and fails the task (0% success). The transformer provides the best trade-off in our runs, improving speed (6.3 vs 6.1), stability (2000 vs 1800) and task success (90% vs 80%) against the RNN baseline. We attribute the failure of non-history scheduler (MLP) to their inability to model temporal learning dynamics in the bin space. Without memory, they cannot identify bin that are useful from non-useful bins, which leads to poor sampling and progression.

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	100 x RNN
Stability (S)	2000	1800	1100
Task Success Rate	90%	80%	0%

† Reported for command only

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.

Multidimensionality ablation (Table IV): Table IV isolates the effect of multidimensionality on transfer by comparing command only, command + DR, and full joint space (command + DR + terrain), i.e. TransCurriculum (Full curriculum). Although the command only curriculum achieves fastest 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 (27% to 18%, nearly 33% reduction). Our findings suggest that multidimensional primarily improves stability and transfer., while history-aware enables high speed training. Most prior work [4], [11], [14] treats velocity, terrain, and domain parameters as independent axis, but in the real-world these dimensions interact and training on a single axis leaves gaps, which show up as sim-to-real transfer loss.

E. Failure Modes and discussion

We identify three recurring failure patterns: (i) lateral deviation on a straight run of 2–8 m; (ii) slips and micro-stumbles on low friction and irregular terrains; and (iii) transient gait irregularities where the robot switches from trot to crawl (0.5–1.0 s) due to contact timing error.

These failures likely arise from (i) imperfect reward shaping focused for heading/straight line tracking, (ii) substrate mismatch, e.g., deformable/loose surfaces (like pebbles, tile, broken rocks) that were not modeled in the simulator, reducing traction and causing slips, and (iii) contact inference failure causing short gait dropout. Despite this, the policy recovers quickly from bumps and slips and maintains functional locomotion across all terrains tested. Addressing these modes through improved terrain modeling, reward shaping, and latent-aware control is a 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 the benefits of prioritizing curriculum expansion over joint space of velocity command, domain parameters, and terrain difficulty that it improves stability and reduces sim-to-real transfer loss (from 27% for command only to 18% for Transcurriculum (full)). In simulation, our approach achieves 6.3 m/s and in zero-shot transfer on Go1 robot it reaches 4.1 ± 0.05 across multiple real-world terrains.

Our evaluations focus our policy on the quadrupedals, and we have not explored a single policy that is transferred across robot morphologies [33], and we do not evaluate bipedal/humanoid locomotion. For future work, the following directions remain promising i) bipedal/humanoid locomotion, ii) morphology generalization, and iii) curriculum to overcome uncertainty or contact related 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] 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.

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

[19] 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.

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

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

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

[33] 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.