Robot Squid Game: Quadrupedal Locomotion for Traversing Narrow Tunnels

Amir Hossain Raj, Dibyendu Das, and Xuesu Xiao

***Abstract*—Quadruped robots demonstrate exceptional potential for navigating complex terrain in critical applications such as search-and-rescue missions and infrastructure inspection. However, autonomous traversal of confined 3D environments—including tunnels, caves, and collapsed structures—remains a significant challenge. Existing methods often struggle with rigid gait patterns, limited adaptability to diverse geometries, and reliance on oversimplified environmental assumptions. This paper introduces a Reinforcement Learning (RL) framework that combines procedural environment generation with policy distillation to enable robust locomotion across various tunnel configurations. Our approach leverages a teacher-student training paradigm, where specialized expert policies trained on procedurally generated tunnel geometries transfer their knowledge to a unified student policy. This strategy eliminates the need for complex reward shaping in end-to-end RL training, simplifying the process by breaking down complicated tasks into smaller, more manageable components that are easier for the robot to learn. By synthesizing diverse tunnel structures during training and distilling navigation strategies into a generalizable policy, our method achieves consistent traversal across complex spatial constraints where conventional approaches fail. We demonstrate, through both simulation and real-world experiments, that our method enables quadruped robots to successfully traverse challenging, confined tunnel environments.**

I. Introduction

The field of legged robotics has witnessed remarkable progress in recent years, with modern quadruped platforms demonstrating unprecedented agility across unstructured outdoor terrain [1]–[3]. (e.g., [4] demonstrating robust adaptation to sand, mud and other terrains). Much research has focused on enabling robots to navigate challenging ground conditions [5]–[7], with approaches ranging from sensorized paws that identify terrain properties [8] to sophisticated control algorithms that maintain stability on uneven surfaces [9]. End-to-end systems using egocentric vision have demonstrated impressive capabilities in traversing stairs, curbs and stepping stones [7], while learning-based methods now enable locomotion across risky terrains with sparse footholds [10]. However, a critical capability gap persists in confined three-dimensional (3D) environments where spatial constraints impose 360° navigation challenges. Such scenarios demand not only ground-level obstacle negotiation but also precise coordination of body posture, limb articulation, and environmental awareness to avoid ceiling collisions and lateral obstructions. Applications ranging from mine shaft inspections to urban disaster response require robots to operate in tunnel-like spaces characterized by irregular cross-sections, tight turns, and limited visual accessibility—environments where current locomotion strategies frequently fail.

All authors are with the Department of Computer Science, George Mason University {araj20, ddas6, xiao}@gmu.edu

Fig. 1: SQUID is deployed in real-world tunnel environments, demonstrating the adaptability and robustness of the proposed approach. The quadrupedal robot relies on limited visual perception to navigate confined spaces, successfully traversing narrow passages and uneven terrain.

Existing methods for navigating confined spaces primarily rely on either geometric planning with static gaits [11] or end-to-end Reinforcement Learning (RL) trained on simplified environmental models [12]. Although hierarchical frameworks that combine classical controllers and learned components appear promising, they encounter three core limitations [13]: (1) oversimplified training that fails to capture real-world structural diversity, (2) highly specialized policies that require substantial retuning for new tunnel shapes, and (3) sensitivity to sensory noise that undermines performance during deployment. Recent work by Buchanan et al. [11] demonstrates body-posture adaptation via a two-layer elevation map but remains constrained to predefined gait patterns, thereby failing to represent more diverse geometries (limitation (1)) and necessitating specialized retuning (limitation (2)). Meanwhile, RL-based approaches [13] can produce dynamic motions in cluttered environments yet remain tied to narrow training setups (limitation (1)) and often degrade under sensor noise (limitation (3)), preventing robust transfer to real-world scenarios.

In this paper, we present SQUID (Skill-fused Quadrupedal locomotion Using Imitation and Distillation), which leverages multi-expert learning and policy distillation [14] to enable robust traversal through confined tunnel environments. SQUID addresses limitations (1), (2), and (3) described above by introducing a privileged learning framework that combines procedural environment generation with policy distillation. Our approach is grounded in two central observations: first, that explicitly modeling the geometric variability of real-world confined 3D spaces (i.e., generating diverse tunnel geometries) is critical for overcoming oversimplified training setups (limitation (1)). Second, decoupling perception-handling strategies from core locomotion is key to robust generalization—hence we initially train specialized "expert" policies with privileged information (focusing on locomotion), and later integrate perception as we transfer their expertise to

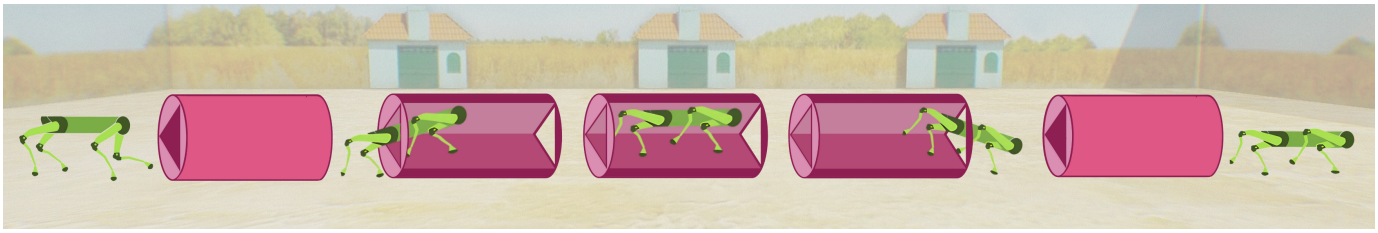

Fig. 2: Quadruped robot executing its learned locomotion policy to traverse a confined tunnel, dynamically adjusting its posture to maintain stability and clearance.

a unified student policy. By doing so, we reduce specialization and the need for policy retuning (limitation (2)) and mitigate sensitivity to noisy real-world sensors (limitation (3)). Unlike prior work that trains a single policy on fixed obstacle distributions, our teacher–student architecture leverages multiple expert policies—each proficient in a distinct tunnel class—and distills their knowledge into a single model through supervised learning. This final policy can robustly handle a wide range of 3D constraints, effectively consolidating the locomotion skills of multiple experts while minimizing further environment-specific tuning.

The contributions of this work are threefold:

1) Tunnel Simulation Pipeline: A procedural generation system creating 3D tunnel environments with parameterized geometric variations (cross-section asymmetry, slope transitions, etc.) that exceed the diversity of existing confined 3D training environments.
2) Privileged Policy Distillation: A teacher–student training paradigm that combines adversarial environment sampling with gradient matching to consolidate specialized expert policies into a single model capable of handling four distinct tunnel classes, thereby minimizing environment-specific retuning.
3) Perceptual Noise Robustness: Experimental validation showing successful real-world deployment using only depth images under partial sensor occlusion and IMU drift.

II. Related Work

In the field of legged robotics, navigating confined spaces presents unique challenges that have been addressed through various methodologies. This section provides an overview of the existing literature, categorized into classical and hierarchical planning approaches, reinforcement learning techniques, privileged learning frameworks, and procedural environment generation methods.

A. Classical and Hierarchical Planning Approaches

Early strategies for confined-space navigation relied on classical planning and optimization techniques. Buchanan et al. [11] introduced perceptive whole-body planning using elevation mapping and motion optimization to adapt robot posture in narrow environments. However, this approach depends on predefined motion primitives, which may not generalize well to irregular geometries. Similarly, Wellhausen et al. [15] developed ArtPlanner, a sampling-based method employing reachability abstraction for legged robots operating in subterranean settings. This method necessitates handcrafted foothold safety heuristics, potentially limiting adaptability in dynamic terrains. Chestnutt et al. [16] proposed global navigation strategies using contact configuration graphs; however, the computational complexity of this approach poses challenges for real-time applications.

B. Reinforcement Learning for Confined-Space Locomotion

Reinforcement Learning (RL) has emerged as a powerful tool for enhancing adaptability in unstructured environments. Xu et al. [13] proposed a hierarchical RL framework that combines classical waypoint planning with low-level policies for 360° obstacle avoidance, achieving successful real-world deployment in vertical shafts. However, reliance on explicit path planning can introduce coordination challenges between hierarchical layers. Miki et al. [17] presented a two-level policy utilizing 3D volumetric representations to navigate under overhangs, though this approach requires separate terrain generators for each environment class. Rudin et al. [18] demonstrated end-to-end RL for dynamic skills such as leaping and crawling; however, their monolithic training framework faced difficulties in sustained confined navigation due to limited state memory. Additionally, approaches like Safe Locomotion within Confined Workspaces [19] using RL have been explored to enhance safety and adaptability in constrained environments. These studies underscore RL's potential but also highlight challenges in consolidating specialized skills across diverse geometries.

C. Privileged Learning and Policy Distillation

Privileged learning frameworks have been instrumental in improving simulation-to-reality transfer by decoupling perception and control. Hwangbo et al. [20] trained teacher policies with full state observability and subsequently distilled these navigation skills into vision-based student policies via behavioral cloning. Lee et al. [3] extended this approach by incorporating curricular hindsight experience replay, facilitating fall recovery and high-speed terrain adaptation. Despite these advancements, single-policy architectures often struggle with conflicting skill requirements in multi-constraint spaces. Recent developments in policy distillation, such as Reinforcement Learning with Demonstrations and Guidance (RLDG), have shown that RL-generated training data can enhance the precision of generalist policies by 40% over human demonstrations [21]. Chebotar et al. [21] demonstrated gradient-based distillation for unified control across manipulation tasks;

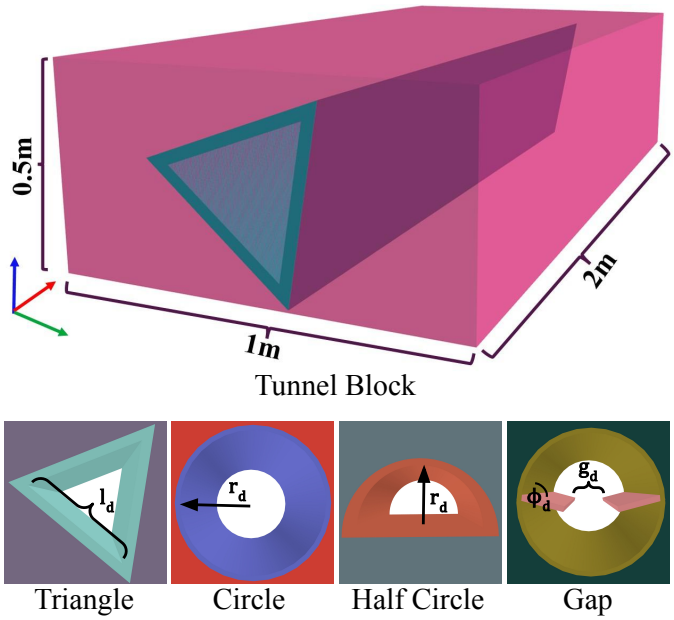

Fig. 3: Simulation training environment for tunnel analysis. The upper portion of the image displays a tunnel block with its measurements and axis orientations. The lower section presents four distinct cross-sectional views of the tunnel, illustrating different structural variations used in the simulation.

however, this approach does not directly address the unique dynamics of legged locomotion in confined volumes.

D. Procedural Environment Generation

Procedural environment generation has become a cornerstone for training robust policies in diverse settings. Miki et al. [12] employed wave function collapse algorithms to synthesize confined spaces with overhangs, enhancing the training diversity for RL policies. Kumar et al. [4] developed automatic terrain difficulty curricula to promote agile locomotion, enabling robots to adapt to varying terrain complexities. Rudin et al. [18] created parkour courses featuring gaps and vertical obstacles; however, their parametric generators lacked the geometric diversity necessary for tunnel-like constraints. Our work advances this paradigm by introducing a physics-aware procedural tunnel generator capable of producing four distinct architectural classes (e.g., triangular, circular) with randomized dimensions, and slope transitions, thereby exceeding the variability present in prior datasets.

III. Method

Our framework combines procedural environment generation, privileged expert training, and vision-based policy distillation to enable robust quadrupedal navigation through confined tunnels. The system architecture progresses through three stages: (1) generating diverse tunnel geometries with parameterized difficulty levels, (2) training specialized expert policies for each tunnel class using privileged simulator information, and (3) distilling multiple experts into a single vision-based student policy through imitation learning.

A. Procedural Tunnel Generation

At the core of our approach is a procedural tunnel generation system designed to address the shortcomings of static and oversimplified training setups. Unlike traditional methods that train RL policies in fixed, predefined environments, our framework dynamically generates diverse tunnel configurations, exposing the robot to a broad range of spatial constraints. This ensures that the learned locomotion strategies remain adaptable rather than overfitting to a single geometry. The transitions in Fig. 2 illustrate how the quadruped dynamically adjusts its posture as it moves through a tunnel, adapting to changes in spatial constraints.

Each tunnel is constructed as a 3D block with a hollowed-out pathway for the robot to traverse (Fig. 3). To systematically vary the spatial constraints, we define four primary tunnel classes, each with tunable difficulty parameters (Table I). These tunnel classes include:

- The equilateral triangle tunnel ($\triangle$) presents a sharp, angular interior that demands careful body rotation and frequent limb adjustments, with the available space shrinking as the edges shorten.
- The full-circle tunnel ($\bigcirc$) offers a uniformly enclosed structure where reducing the radius progressively increases the difficulty, forcing the robot to crouch or adjust its gait dynamically.
- The half-circle tunnel ($\cap$) introduces additional complexity by randomly flipping its Z-normal, requiring adaptation to inverted terrain.
- The gap tunnel ($\vdash\dashv$) features a central void with elevated side shelves, posing a challenge for stable foothold selection, especially as the gap widens or the shelf angles increase.

Although these geometric primitives appear simple, they are highly expressive in capturing fundamental locomotion challenges. Real-world confined spaces often exhibit sharp angles, uneven terrain, varying clearance, and rotational asymmetries, all of which are represented in our tunnel classes. Additionally, the procedural nature of our framework prevents the learning process from being constrained to fixed obstacle distributions. Key environmental factors such as tunnel width, height, curvature, and inclination vary continuously across training episodes, ensuring a rich distribution of training data. By randomly rotating certain tunnels, flipping their orientations, and altering their connectivity, we force the robot to develop generalizable motion strategies rather than memorizing specific paths.

To further improve adaptability, our tunnel configurations are sequentially connected during training. The robot begins in simpler environments and gradually progresses to more complex ones, encountering increasingly tight, irregular, or asymmetric spaces. This curriculum-based approach ensures that the policy learns effective locomotion strategies incrementally, reinforcing fundamental movement principles before tackling highly constrained navigation.

Tunnel Class	Cross-section Geometry	Difficulty Parameters	Generation Method
Triangle △	Three equal sides	Edge length $l_d = l_0 - 0.1d$, Rotation $\theta \sim U(0, 360°)$	Random rotation per segment
Circle ◯	Closed circular profile	Radius $r_d = r_0 - 0.05d$	Fixed orientation
Half-Circle ⋂	Semicircular arc	Radius $r_d = r_0 - 0.07d$, Z-normal $\in \{-1, +1\}$ (random)	Flipped Z-normal
Gap ⊢⊣	Central gap with side shelves	Gap width $g_d = g_0 + 0.2d$, Shelf angle $\phi_d = \phi_0 + 5°d$	Symmetric shelves

TABLE I: Tunnel geometry specifications, detailing the cross-section geometry, difficulty parameters, and generation methods used for each tunnel class.

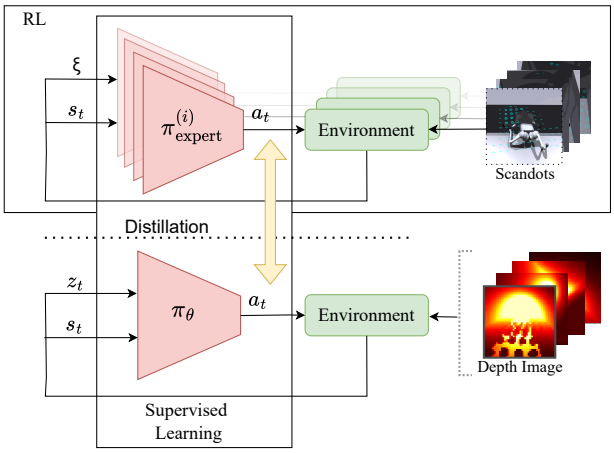

Fig. 4: Training pipeline for SQUID. Teacher policies are first trained using RL with privileged information for each tunnel class. Distillation transfers expert knowledge to a unified student policy, which is trained using onboard sensing.

By synthesizing a wide variety of tunnel structures and exposing the robot to continually changing spatial constraints, our method avoids the common pitfalls of static training environments. The result is a locomotion policy that retains the flexibility and robustness needed for real-world deployment, effectively bridging the gap between structured simulation training and unpredictable, confined 3D spaces.

B. Privileged Teacher Policies

The teacher policies are trained using RL framework, mapping observations to actions that enable the robot to navigate tunnel environments of varying difficulty. Each teacher policy is trained independently within its assigned tunnel class, ensuring specialization in handling the unique constraints of that environment. Given that we define four tunnel classes—triangle (△), circle (◯), half-circle (⋂), and gap tunnel (⊢⊣)—we train four expert policies, one for each tunnel class. These experts leverage privileged state information during training to learn locomotion strategies, which are later distilled into a single deployable student policy. To be specific, the i-th expert policy is denoted as:

$$\pi^{(i)}_{\text{expert}}(a_t | s_t, \xi),$$

where, s_t represents the state at time t, including proprioceptive and exteroceptive observations; ξ denotes privileged simulator information, which includes ground-truth environmental details that are unavailable during student policy execution; and a_t is the action taken at time t.

1) Observation Space: The observation space consists of proprioceptive and exteroceptive measurements that provide comprehensive information about the robot's motion and surroundings. Specifically, in the observation space, base linear and angular velocities $(\mathbf{v}_b, \boldsymbol{\omega}_b)$ capture the robot's movement dynamics; gravity vector orientation provides information about the robot's pose relative to gravity; joint positions and velocities $(\mathbf{q}_j, \dot{\mathbf{q}}_j)$ track the robot's joint configurations; previous actions maintain temporal consistency in decision-making; and terrain measurement is a 108-dimensional grid around the robot's base, encoding distances from the terrain surface to the robot's body height. By incorporating both proprioceptive and terrain-based sensory inputs, the teacher policies have access to high-fidelity state information, enabling them to learn robust locomotion strategies tailored to their specific tunnel class.

2) Action Space: Each expert policy outputs a 12-dimensional action vector, corresponding to desired joint positions for the 12 motors (three per leg). These actions are passed through a Proportional-Derivative (PD) controller, which converts them into motor torques for actuation:

$$\boldsymbol{\tau}_j = k_p(\mathbf{q}^d_j - \mathbf{q}_j) + k_d(\dot{\mathbf{q}}^d_j - \dot{\mathbf{q}}_j),$$

where $\mathbf{q}^d_j$ and $\dot{\mathbf{q}}^d_j$ are the desired joint positions and velocities given by the teacher policy, $\mathbf{q}_j$ and $\dot{\mathbf{q}}_j$ are the current joint positions and velocities, and k_p and k_d are proportional and derivative gains controlling the system stiffness and damping. Using joint position control instead of direct torque control ensures stable learning and efficient locomotion, as the system does not need to model complex actuator dynamics explicitly.

3) Reward Function: The reward function encourages efficient, stable, and collision-free locomotion while minimizing energy consumption. It is formulated as a weighted sum of individual reward terms (Table II), where each term reinforces a specific desirable behavior.

To ensure precise trajectory tracking, we include linear velocity tracking (r_{lv}) and angular velocity tracking (r_{av}), which encourage the robot to match a target translational velocity $\mathbf{v}^*_{b,xy}$ and yaw rate $\omega^*_{b,z}$, respectively. These terms

penalize deviations from the desired motion commands. To maintain stable locomotion, vertical velocity (r_{vp}) and horizontal angular velocity penalties (r_{ap}) discourage excessive fluctuations in body movement. Additionally, joint motion (r_{jm}) and torque penalties (r_{τ}) ensure smooth and efficient actuation by penalizing high joint accelerations, velocities, and large torque outputs. Collision avoidance is enforced through a collision penalty (r_{coll}), which assigns negative rewards for contacts with tunnel walls, encouraging safer navigation through constrained spaces. Finally, step duration reward (r_{step}) is introduced to promote structured footstep timing. It is based on the air time $t_{\text{air},f}$ of each leg f, ensuring a balance between stance and swing phases. A reference duration of 0.5 is used to encourage stable and efficient gaits. This reward structure ensures that expert policies learn collision-free, energy-efficient, and dynamically stable locomotion strategies while generalizing across a variety of tunnel environments.

Reward Term	**Equation**	**Description**
r_{lv}	$\exp\left(-\frac{\|\mathbf{v}^*_{b,xy}-\mathbf{v}_{b,xy}\|^2}{0.25}\right)$	Linear velocity tracking
r_{av}	$\exp\left(-\frac{\|\omega^*_{b,z}-\omega_{b,z}\|^2}{0.25}\right)$	Angular velocity tracking
r_{vp}	$-v^2_{b,z}$	Vertical velocity penalty
r_{ap}	$-\|\omega_{b,xy}\|^2$	Horizontal angular velocity penalty
r_{jm}	$-\|\ddot{\mathbf{q}}_j\|^2 - \|\dot{\mathbf{q}}_j\|^2$	Joint motion penalty
r_{τ}	$-\|\boldsymbol{\tau}_j\|^2$	Joint torque penalty
r_{coll}	$-n_{\text{collision}}$	Collision penalty
r_{step}	$\sum_{f=1}^{4}\left(t_{\text{air},f}-0.5\right)$	Step duration reward

TABLE II: Reward Terms for Privileged Teacher Policies

C. Student Policy Using Distillation

Once the four expert policies ($\pi^{(i)}_{\text{expert}}$) are trained, we employ a policy distillation framework to consolidate their knowledge into a single vision-based student policy (Fig. 4). Unlike the experts, which rely on privileged simulator information, the student policy learns to navigate using depth images and historical proprioception, making it suitable for real-world deployment.

To achieve this, we use Dataset Aggregation (DAgger) [22], an iterative imitation learning approach that mitigates distribution shift by collecting on-policy rollouts under the student policy while receiving corrective feedback from the teacher policies. The student policy is denoted as:

$$\pi_\theta(a_t|s_t, z_t), \tag{1}$$

where s_t is the current state, consisting of proprioceptive inputs; z_t represents the depth image encoding environmental obstacles; and a_t is the action controlling the robot's motion.

The distillation objective is to minimize the discrepancy between the student and expert actions, formulated as:

$$L(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{i=1}^{4}\ell\left(\pi^{(i)}_{\text{expert}}(s_t,\xi), \pi_\theta(s_t, z_t)\right)\right], \tag{2}$$

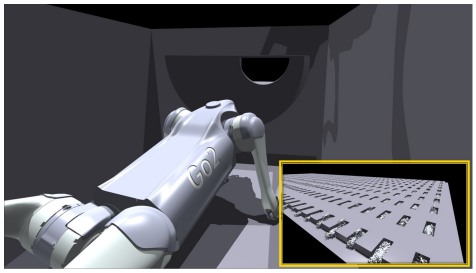

Fig. 5: Parallelized training of quadrupedal robots in confined tunnels (inset), with a zoomed-in view of a single environment.

where $\ell(\cdot)$ is a loss function (e.g., Mean Squared Error or Cross-Entropy Loss) that quantifies the alignment between expert and student actions, and τ represents the trajectory distribution under the student policy. This gradient-based distillation process enables the student policy to inherit robust locomotion behaviors from the teachers across all tunnel classes without requiring privileged information.

IV. Experimental Results

In this section, we present an extensive evaluation of SQUID through simulation-based trials and real-world deployments. We first outline the experimental setup and performance metrics. We then compare our method against several baselines, conduct ablation studies to understand the contribution of each component, and finally discuss real-world test results.

A. Experimental Setup and Metrics

1) Robot Platform and Simulation: We use a simulated Unitree Go2 quadruped robot configured with 12 actuated joints. All simulation experiments are conducted using Isaac Gym [23], allowing parallelized training and testing across multiple tunnel environments. The robot's onboard sensor suite in the simulation includes forward-facing depth camera providing 64×48 pixel images; an Inertial Measurement Unit for orientation estimates; and joint encoders for proprioceptive feedback. We train and evaluate on four tunnel classes with increasing difficulty levels controlled by parameters in Table I and values in Table III. Training is conducted using massive parallelization (Fig 5) for each tunnel class where the columns are generated as a chain of tunnel blocks, with increasing difficulty, connected via narrow passages.

2) Implementation and Training: The teacher policies utilize a multi-layer perceptron architecture with tanh activation functions. These policies are trained using Proximal Policy Optimization (PPO) [24]. For the student policy, we employ a convolutional neural network [25] encoder to process depth images, followed by a Gated Recurrent Unit [26] to maintain temporal context across frames and proprioceptive history. The student network's final layers map encoded features to the 12-dimensional joint position action space. All training is conducted on an NVIDIA RTX 3090 GPU, enabling parallelization of 2048 environments in Isaac Gym. The teacher policies converge after approximately 10,000 iterations (∼8 hours of training time per tunnel class). During distillation,

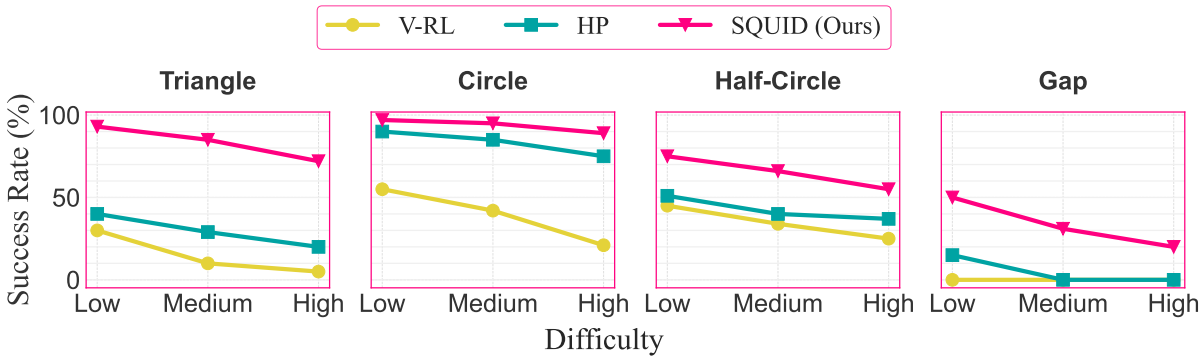

Fig. 6: Success Rate comparison across different tunnel classes and difficulty levels. Each plot represents a specific tunnel geometry, showing the performance of various methods (V-RL, HP, and SQUID) as the difficulty increases. The SQUID policy consistently outperforms baselines, particularly in constrained environments like triangular and gap tunnels.

depth images are augmented with random noise and occasional dropout to improve robustness against sensor imperfections in real-world deployment.

3) Performance Metrics: We adopt four metrics reported as average over 50 trials per tunnel type and difficulty level: Success Rate is the percentage of trials in which the robot successfully reaches the tunnel exit without collisions that cause a reset; Trajectory Completion Time is the average time taken to traverse a tunnel segment; Collision Frequency is the number of body collisions with tunnel walls, measured per meter traveled; and Energy Consumption is the sum of the absolute motor torques over a traversal, normalized by distance.

Tunnel Class	Low	Medium	High
Triangle △	$l_d = [0.42m, 0.48m]$ $\theta \sim U(0, 90°)$	$l_d = [0.35m, 0.42m]$ $\theta \sim U(0, 180°)$	$l_d = [0.30m, 0.35m]$ $\theta \sim U(0, 360°)$
Circle ○	$r_d = [0.22m, 0.24m]$	$r_d = [0.16m, 0.22m]$	$r_d = [0.12m, 0.18m]$
Half-Circle ⋂	$r_d = [0.40m, 0.42m]$	$r_d = [0.35m, 0.40m]$	$r_d = [0.25m, 0.32m]$
Gap ⊢⊣	$g_d = [0.1m, 0.2m]$ $\phi_d = [0°, 5°]$	$g_d = [0.2m, 0.35m]$ $\phi_d = [5°, 10°]$	$g_d = [0.35m, 0.4m]$ $\phi_d = [10°, 15°]$

TABLE III: Difficulty Parameters for Tunnel Classes.

B. Comparison with Baselines

We compare SQUID against two baselines in simulation using difficulty parameters from Table III. Vanilla RL (V-RL) is a single PPO agent trained end-to-end on all tunnel classes simultaneously with similar observation space as teacher policies. Hierarchical Planner (HP) [13] is a two-layer system where a high-level planner generates waypoints and a low-level controller executes footstep motions. The planner uses elevation mapping for obstacle detection.

1) Success Rate: Fig. 6 presents the Success Rates across tunnel classes and difficulty levels. SQUID achieves the highest Success Rates across all configurations with the most noticeable advantage in triangular and gap tunnels, where constrained navigation demands precise body articulation. HP performs well in structured tunnels (like Circle) but struggles in environments requiring adaptive motion strategies (like Gap). V-RL suffers from poor generalization, failing frequently in complex geometries and high difficulty levels.

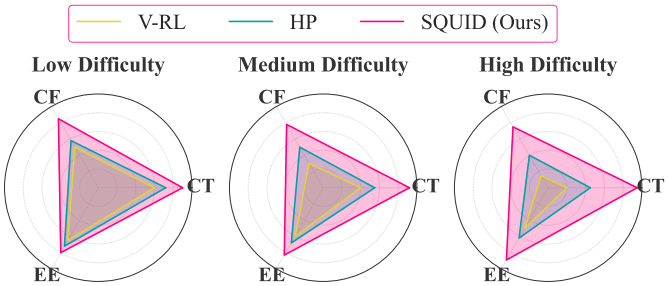

Fig. 7: Comparison of Completion Time (CT), Collision Frequency (CF), and Energy Efficiency (EE) across difficulty levels. SQUID outperforms V-RL and HP, maintaining faster traversal, lower collisions, and better energy efficiency, with increasing advantages in higher difficulty tunnels.

2) Traversal Efficiency, Collision Avoidance, and Energy Consumption: The radar plots in Fig. 7 summarize three key performance metrics—Completion Time, Collision Frequency, and Energy Consumption—across low, medium, and high difficulty levels. These three interdependent aspects directly impact the efficiency and robustness of the locomotion policy. To maintain a consistent interpretation where higher values indicate better performance, these metrics have been inverted in the plot.

Completion Time: SQUID consistently achieves faster traversal speeds compared to HP, which tends to be overly cautious. While HP ensures stability, it sacrifices speed, leading to long Completion Time. V-RL occasionally stalls in tight spaces, further increasing traversal time. SQUID strikes an effective balance, maintaining fast but stable motion across varying tunnel geometries.

Collision Frequency: SQUID exhibits the lowest collision rates, benefiting from its multi-expert knowledge transfer. It effectively maintains clearance from tunnel walls while adapting dynamically to asymmetric structures. HP struggles in tunnels requiring whole-body posture adaptation, while V-RL frequently collides due to erratic foot placements and inadequate control in constrained spaces.

Energy Consumption: SQUID achieves greater energy efficiency by ensuring smoother gait transitions and reducing unnecessary corrective movements. HP, although stable, expends more energy due to its slow traversal, which increases total energy expenditure. V-RL is the least efficient, consuming excessive energy due to frequent stops, unstable gaits, and inefficient motor commands.

C. Ablation Studies

To systematically assess the contribution of key components in our SQUID framework, we conduct a series of ablation experiments, selectively modifying critical elements and evaluating their impact on policy performance. The results, summarized in Table IV, report the Success Rates under different ablation settings. These experiments provide insights into the importance of multi-expert learning, procedural generation, privileged perception, and structured reward shaping.

1) Single Teacher vs. Multiple Teachers: To test the necessity of multiple expert policies, we train a student policy using only a single teacher, specifically the expert trained in circular tunnels. The resulting policy exhibits moderate success in environments similar to the training distribution but fails to generalize to tunnels with asymmetric constraints, such as triangular or gap tunnels. This highlights that policies trained with a single teacher overfit to specific tunnel geometries, leading to poor adaptability when encountering new structural variations. In contrast, our multi-teacher approach, where each expert specializes in a distinct tunnel class, provids a more diverse knowledge base. The distilled student learns adaptive behaviors across varying tunnel configurations, leading to more consistent success across all settings.

2) Procedural Generation Disabled: To examine the role of environmental diversity, we train a student policy in a fixed, non-procedural environment where tunnel shapes remain constant across training episodes. The policy achieves reasonable success in familiar scenarios but fails in tunnels with unexpected variations in curvature, slope, or orientation. Without exposure to procedural variations during training, the policy becomes rigid, adapting poorly to real-world variations. This confirms that procedural generation plays a crucial role in promoting generalization by exposing the policy to a wide range of environmental constraints.

3) Two-Layer Elevation Map vs. Ground Elevation Map: We train teacher policies using a two-layer elevation map that concatenates floor and ceiling elevations and compare them to policies trained with only ground elevation maps. Policies using the two-layer elevation representation fail to converge, as the additional ceiling constraints introduce conflicting optimization objectives, leading to unstable body posture adjustments and frequent stalls. In contrast, policies trained with only ground elevation maps successfully learn stable locomotion strategies, achieving better generalization and traversal efficiency. These results indicate that explicitly encoding ceiling constraints increases learning complexity without improving policy performance, suggesting that alternative ceiling-aware representations should be explored.

Because teacher policies are trained in separate tunnel classes, they can estimate ceiling elevation using only the ground heightmap and adjust posture accordingly. They rely on privileged information to estimate elevation changes, ensuring smooth transitions at tunnel entries and exits. However, distillation with ground elevation alone does not generalize, requiring the student policy to use a depth map for exteroception.

4) Reward Shaping Simplifications: We further analyze how structured reward functions contribute to stable locomotion by removing key components from the expert training stage. Eliminating the collision penalty results in erratic movement patterns, with the policy frequently making contact with tunnel walls and ceilings due to the absence of a strong deterrent against risky postures. Similarly, removing the vertical velocity penalty leads to an increase in destabilizing hopping behaviors, particularly in environments with variable elevation. These behaviors compromise stability and traversal efficiency, underscoring the importance of structured reward shaping for safe and effective locomotion.

Ablation Setting	**Success Rate**
Full SQUID (Ours)	**70%**
Single Teacher	35%
No Procedural Generation	39%
Two-Layer Elevation Map	19%

TABLE IV: Success Rates (%) for different ablation settings, demonstrating the impact of key SQUID components.

D. Real-World Deployment

To assess the sim-to-real transferability of our SQUID policy, we deploy the trained model on a Unitree Go2 quadruped in a controlled tunnel environment (Fig. 1). The real-world setup consists of three 1-meter tunnel segments with circle, half-circle, and triangular cross-sections, constructed using plywood barriers and curved PVC enclosures. The tunnels are designed to replicate the constrained geometries encountered in simulation, minimizing the sim-to-real transfer gap.

The quadruped is equipped with a 4D LiDAR, which generates depth images that are processed through an encoder network before being passed to the deployable SQUID policy. Unlike simulation, where the robot has access to precise state information, the real-world deployment introduces sensor noise and depth artifacts, providing a challenging test for generalization.

The robot successfully navigates the tunnel environments while adapting to real-world inconsistencies. Minor deviations in tunnel structure and occasional sensor occlusions lead to variations in locomotion strategies, requiring the policy to make real-time adjustments. Despite these challenges, the robot maintains stable traversal and completes tunnel passages without requiring manual intervention. Future improvements can focus on refining perception models to better handle sensor noise and occlusions, further enhancing robustness in real-world conditions.

V. Conclusions and Future work

This paper presents SQUID, a RL framework combining procedural environment generation and privileged policy distillation to achieve robust quadrupedal locomotion in confined 3D tunnel environments. SQUID leverages multiple expert teacher policies trained on diverse procedurally generated tunnel geometries and distills their specialized knowledge into a unified vision-based student policy, effectively addressing limitations of existing methods such as overspecialization, sensitivity to sensor noise, and reliance on simplified environmental assumptions. Experimental results demonstrate that SQUID consistently outperforms baseline approaches across various tunnel geometries and difficulty levels, achieving higher success rates, faster traversal times, fewer collisions, and improved energy efficiency. Real-world deployment further validates the robustness of SQUID under realistic sensor conditions. Future work includes enhancing perception robustness through advanced sensor fusion techniques, integrating online terrain reconstruction for dynamic adaptation, exploring multi-robot coordination within confined spaces, and generalizing the procedural generation pipeline to more complex subterranean environments.

References

[1] M. Sorokin, J. Tan, C. K. Liu, and S. Ha, "Learning to navigate sidewalks in outdoor environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3906–3913, 2022.

[2] K. Caluwaerts, A. Iscen, J. C. Kew, W. Yu, T. Zhang, D. Freeman, K.-H. Lee, L. Lee, S. Saliceti, V. Zhuang, N. Batchelor, S. Bohez, F. Casarini, J. E. Chen, O. Cortes, E. Coumans, A. Dostmohamed, G. Dulac-Arnold, A. Escontrela, E. Frey, R. Hafner, D. Jain, B. Jyenis, Y. Kuang, E. Lee, L. Luu, O. Nachum, K. Oslund, J. Powell, D. Reyes, F. Romano, F. Sadeghi, R. Sloat, B. Tabanpour, D. Zheng, M. Neunert, R. Hadsell, N. Heess, F. Nori, J. Seto, C. Parada, V. Sindhwani, V. Vanhoucke, and J. Tan, "Barkour: Benchmarking animal-level agility with quadruped robots," 2023. [Online]. Available: https://arxiv.org/abs/2305.14654

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

[4] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," in *Robotics: Science and Systems*, 2021.

[5] H. Duan, A. Malik, M. S. Gadde, J. Dao, A. Fern, and J. Hurst, "Learning dynamic bipedal walking across stepping stones," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 6746–6752.

[6] H. Duan, B. Pandit, M. S. Gadde, B. Van Marum, J. Dao, C. Kim, and A. Fern, "Learning vision-based bipedal locomotion for challenging terrain," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 56–62.

[7] A. Agarwal, A. Kumar, J. Malik, and D. Pathak, "Legged locomotion in challenging terrains using egocentric vision," in *Conference on Robot Learning (CoRL)*, 2022.

[8] A. Vangen, T. Barnwal, J. A. Olsen, and K. Alexis, "Terrain recognition and contact force estimation through a sensorized paw for legged robots," *arXiv preprint arXiv:2311.03855*, 2023.

[9] H. Zhu, D. Wang, N. Boyd, Z. Zhou, L. Ruan, A. Zhang, N. Ding, Y. Zhao, and J. Luo, "Terrain-perception-free quadrupedal spinning locomotion on versatile terrains: Modeling, analysis, and experimental validation," *Frontiers in Robotics and AI*, vol. 8, Oct. 2021.

[10] R. Yu, Q. Wang, Y. Wang, Z. Wang, J. Wu, and Q. Zhu, "Walking with terrain reconstruction: Learning to traverse risky sparse footholds," *arXiv preprint arXiv:2409.15692*, 2024.

[11] R. Buchanan, T. Bandyopadhyay, M. Bjelonic, L. Wellhausen, M. Hutter, and N. Kottege, "Walking posture adaptation for legged robot navigation in confined spaces," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2148–2155, 2019.

[12] T. Miki, J. Lee, L. Wellhausen, and M. Hutter, "Learning to walk in confined spaces using 3d representation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

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

[14] A. A. Rusu, S. G. Colmenarejo, Çaglar Gülçehre, G. Desjardins, J. Kirkpatrick, R. Pascanu, V. Mnih, K. Kavukcuoglu, and R. Hadsell, "Policy distillation," *CoRR*, vol. abs/1511.06295, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:1923568

[15] L. Wellhausen and M. Hutter, "Artplanner: Robust legged robot navigation in the field," *arXiv preprint arXiv:2303.01420*, 2023.

[16] J. Chestnutt, J. Kuffner, K. Nishiwaki, S. Kagami, K. Kaneko, M. Fukushi, K. Nagasaka, M. Inaba, and H. Inoue, "Global planning methods for legged robots on rough terrain," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 1245–1252.

[17] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[18] N. Rudin, D. Hoeller, L. Wellhausen, and M. Hutter, "Learning to perform dynamic legged manoeuvres on flipper steps: A parkour approach," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6789–6796, 2022.

[19] T. He, C. Zhang, W. Xiao, G. He, C. Liu, and G. Shi, "Agile but safe: Learning collision-free high-speed legged locomotion," in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[20] 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.

[21] Y. Chebotar, K. Hausman, Y. Lu, T. Xiao, D. Kalashnikov, J. Varley, A. Irpan, P. Pastor, C. Finn, and S. Levine, "Reinforcement learning with demonstrations and guidance: A unified framework for robotic manipulation," in *Proceedings of the 2021 Conference on Robot Learning*, 2021, pp. 1309–1318.

[22] S. Ross, G. J. Gordon, and J. A. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, vol. 15. PMLR, 2011, pp. 627–635. [Online]. Available: https://proceedings.mlr.press/v15/ross11a.html

[23] 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," in *NeurIPS 2021 Track Datasets and Benchmarks*, 2021.

[24] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70. PMLR, 2017, pp. 3057–3065.

[25] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[26] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations (ICLR)*, 2015.