ADEPT: Adaptive Diffusion Environment for Policy Transfer Sim-to-Real

Youwei Yu, Junhong Xu, and Lantao Liu
Indiana University Bloomington, USA
{youwyu, xu14, lantao}@iu.edu

***Abstract*—Model-free reinforcement learning has emerged as a powerful method for developing robust robot control policies capable of navigating through complex and unstructured environments. The effectiveness of these methods hinges on two essential elements: (1) the use of massively parallel physics simulations to expedite policy training, and (2) an environment generator tasked with crafting sufficiently challenging yet attainable environments to facilitate continuous policy improvement. Existing methods of outdoor environment generation often rely on heuristics constrained by a set of parameters, limiting the diversity and realism. In this work, we introduce ADEPT, a novel Adaptive Diffusion Environment for Policy Transfer in the zero-shot sim-to-real fashion that leverages Denoising Diffusion Probabilistic Models to dynamically expand existing training environments by adding more diverse and complex environments adaptive to the current policy. ADEPT guides the diffusion model's generation process through initial noise optimization, blending noise-corrupted environments from existing training environments weighted by the policy's performance in each corresponding environment. By manipulating the noise corruption level, ADEPT seamlessly transitions between generating similar environments for policy fine-tuning and novel ones to expand training diversity. To benchmark ADEPT in off-road navigation, we propose a fast and effective multi-layer map representation for wild environment generation. Our experiments show that the policy trained by ADEPT outperforms both procedural generated and natural environments, along with popular navigation methods.**

I. Introduction

Autonomous navigation across unstructured complex environments necessitates the development of control policies that exhibit both robustness and smooth interactions within these challenging environments [31, 44, 74]. In this work, we target the training of a control policy that allows robots to adeptly navigate through diverse environments, such as unstructured indoor-outdoor environments and complex off-road terrains.

Recent advancements in reinforcement learning (RL) have shown great promise in enhancing autonomous robot navigation in challenging scenarios [71, 45, 1]. While an ideal case involves training an RL policy to operate seamlessly in all possible environments, the complexity of real-world scenarios makes it impractical to enumerate the entire spectrum of possibilities. Popular methods, including curriculum learning in simulation [34] and fine-tuning in real world [62], and imitation learning using real-world collected data [73] encounter limitations in terms of training data diversity and the human efforts required. Recently, the real-to-sim-to-real paradigm [13] features multi-modal information through radiance field rendering [36] real-world environments in the simulation. However, without sufficient data and training, the application of learned policies to dissimilar scenarios becomes challenging, thereby hindering efforts to bridge the out-of-distribution gap. Additionally, existing solutions, such as traversability estimation [53, 64] for motion sampling [18, 72] and optimization methods [59, 79], may exhibit fragility due to sensor noise and complex characteristics of vehicle-terrain interactions.

To tackle this challenge, we propose ADEPT, an Adaptive Diffusion Environment generator for Policy Transfer in the zero-shot sim-to-real fashion. ADEPT is designed to co-evolve with the policy, producing new environments that effectively push the boundaries of the policy's capabilities. Starting with an initial environment dataset, which may be from existing data or environments generated by generative models, ADEPT is capable of expanding it into new and diverse environments. The significant contributions include:

- **Adjustable Generation Difficulty:** ADEPT dynamically modulates the complexity of generated environments by optimizing the initial noise (latent variable) of the diffusion model. It blends noise-corrupted environments from the training environments, guided by weights derived from the current policy's performance. As a result, the reverse diffusion process, starting at the optimized initial noise, can synthesize environments that offer the right level of challenge tailored to the policy's current capabilities.
- **Adjustable Generation Diversity:** By adjusting the initial noise level before executing the Denoising Diffusion Probabilistic Model (DDPM) reverse process, ADEPT effectively varies between generating challenging environments and introducing new environment geometries. This capability is tailored according to the diversity present in the existing training dataset, enriching training environments as needed throughout the training process. Such diversity is crucial to ensure the trained policy to adapt and perform well in a range of previously unseen scenarios.

We specifically target the training of adept navigation through diverse off-road terrains, such as ones characterized by varying elevations, irregular surfaces, and obstacles. This article extends our previous work [77] from multiple perspectives:

- **Scalable Generation:** Our ADEPT focuses on expos-

ing agents to contiguous environments across successive training epochs. Unlike discontinuous environments suited for local planning or super large environments that incur computation burdens, this approach enhances performance to long-horizon tasks with efficiency.

- **Off-Road Environment Representation:** Rather than bare terrain elevations, we extend environments as multi-layer maps, from the terrain elevation to the surface canopy, offering effective generation of elevations and plants compared to direct fine geometry inference.
- **Stereo-Vision Perception Simulation:** For the key attribute, perception domain, we simulate the depth measurement noise from simulator-rendered infrared stereo images with stereo matching. Instead of overly complicate hand-crafted noise models to the perception (e.g., depth image or elevation map), we randomize the single infrared noise model which offers simple controllability and realism.

We systematically validate the proposed ADEPT framework by comparing it with established environment generation methods [40, 45] for training navigation policies on uneven terrains. Our experimental results indicate that ADEPT offers enhanced generalization capabilities and faster convergence. Building on this core algorithm, we integrate ADEPT with teacher-student distillation [9] and domain randomization [3] in physics and perception. We evaluate the distilled student policy with zero-shot transfer to simulation and real-world experiments. The results reveal our framework's superiority over competing methods [78, 72, 71, 57] in key performance metrics.

II. Related Work

A. Navigation in the Wild

Navigation in unstructured outdoor environments requires planners to handle more than simple planar motions. Simulating full terra-dynamics for complex, deformable surfaces like sand, mud, and snow is computationally intensive. Consequently, most model-based planners use simplified kinematics models for planning over uneven terrains [74, 67, 75, 39, 49] and incorporate semantic cost maps to evaluate traversability not accounted in the simplified model [44, 63, 19, 53, 64]. Continuously learning the semantic traversability is powerful as it can incorporate multi-modal information so aim to offer a plug-and-play solution that can seamlessly integrate into the state-of-the-art semantic learning methods. Our method can follow waypoints optimized on the traversability map. Imitation learning (IL) methods [73, 51, 61] bypass terrain modeling by learning from expert demonstrations but require labor-intensive data collection. On the other hand, model-free RL does not require expert data and has shown impressive results enabling wheeled [34, 28, 71, 54] and legged robots [40, 45, 30, 46] traversing uneven terrains by training policies over diverse terrain geometries. However, the challenge is to generate realistic environments to bridge the sim-to-real gap. The commonly-used procedural generation methods [45, 40] are limited by parameterization and may not accurately reflect real-world environment geometries. Our work addresses this by guiding a diffusion model trained on natural environments to generate suitable off-road environments for training RL policies.

B. Sim-to-Real Robot Learning

[40] proposed zero-shot sim-to-real quadruped locomotion where a Temporal Convolutional Network (TCN) encodes the state-action history to reconstruct the privileged information. To leverage exteroceptive information for additional reconstruction, [45] proposed the belief encoder-decoder module that enables robust behavior even with perception occlusions. Subsequently, [30] proposed a compact and robust system where the high-level classic path planner guides the low-level learned controller to achieve superior successes. However, these works have restrictions on the procedural generation terrain diversity. On one hand, to learn in the real world to unseen scenarios, RMA [37] distilled a parkour policy on a latent space of environment extrinsic from the state-action history. But it cannot distill multiple specialized skill policies into one parkour policy [81]. On the other hand, three-dimensional procedural environment generation [46] could empower locomotion in confined spaces, with limits in realism. [81] proposed soft and hard obstacle constraints for smooth skill learning, while the environment is still restricted by human-crafted stairs and boxes.

C. Automatic Curriculum Learning and Controllable Generation

Our method is a form of automatic curriculum learning [56, 50], where it constructs increasingly challenging environments to train RL policies. While one primary goal of curriculum learning in RL is to expedite training efficiency [27, 14, 16], recent work shows that such automatic curriculum can be a by-product of unsupervised environment design (UED) [12, 69, 70, 33, 41]. It aims to co-evolve the policy and an environment generator during training to achieve zero-shot transfer during deployment. Unlike prior works in UED, the environments generated by our method are grounded in realistic environment distribution learned by a diffusion model and guided by policy performance. Recently, a concurrent work proposes Grounded Curriculum Learning [68]. It uses a variational auto-encoder (VAE) to learn realistic tasks and co-evolve a parameterized teacher policy to control VAE-generated tasks using UED-style training. In contrast, our work uses a sampling-based optimization method to control the diffusion model's initial noise for guided generation.

Controllable generation aims to guide a pre-trained diffusion model to generate samples that are not only realistic but also satisfy specific criteria. A commonly used strategy is adding guided perturbations to modify the generation process of a pre-trained diffusion model using scores from the conditional diffusion [24, 2] or gradients of cost functions [80]. Another approach is to directly optimize the weights of a pre-trained diffusion model so that the generated samples optimize some

objective function. By treating the diffusion generation process as a Markov Decision Process, model-free reinforcement learning has been used to fine-tune the weights of a pre-trained diffusion model [7, 65]. This approach can also be viewed as sampling from an un-normalized distribution, given a pre-trained diffusion model as a prior [66]. Our work is closely related to initial noise optimization techniques for guiding diffusion models [5, 35, 22]. Instead of refining the diffusion model directly, these methods focus on optimizing the initial noise input. By freezing the pre-trained diffusion model, we ensure that the generated samples remain consistent with the original data distribution. In contrast to existing approaches focusing on content generation, our work integrates reinforcement learning (RL) with guided diffusion to train generalizable robotic policies.

III. Preliminaries

A. Problem Formulation

We represent the environment as e and a common practice is a multi-channel discretized map, denoted as $e \in \mathbb{R}^{C\times W\times H}$, where C, W and H represent the number of channels, width, and height, respectively. Similar to most works in training RL policies for zero-shot sim-to-real navigation [81, 26], we use the high-performance physics simulator [42] to model the state transitions of the robot moving in environments $s_{t+1} \sim p(s_{t+1}|s_t, a_t, e)$. Here, $s \in \mathcal{S}$ and $a \in \mathcal{A}$ represent the robot's state and action, and each realization of e specifies a unique environment. An optimal policy $\pi(a|s, e; \theta)$ can be found by maximizing the expected cumulative discounted reward. Formally,

$$\theta^* = \arg\max_{\theta} \mathbb{E}_{\substack{a_t\sim\pi(a_t|s_t,e), s_0\sim p(s_0),\\ e\sim p(e), s_{t+1}\sim p(s_{t+1}|s_t,a_t,e)}} \left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t)\right], \tag{1}$$

where $p(s_0)$ is the initial state distribution and $p(e)$ denotes the distribution over the environments. Due to the environment e imposing constraints on the robot's movement, the policy optimized through Eq. (1) is inherently capable of avoiding hazards on convex surfaces and among diverse objects. We aim to dynamically evolve the environment distribution $p(e)$ based on the policy's performance, ensuring training efficiency and generating realistic environments.

B. Adaptive Curriculum Reinforcement Learning for Environment-Aware Policy Optimization

A theoretically correct but impractical solution to Eq. (1) is to train on all possible environments $\Lambda = (e^1, ..., e^\infty)$, with $p(e)$ as a uniform distribution over Λ. However, the vast variability of environment geometries makes this infeasible. Even if possible, it might produce excessively challenging or overly simple environments, risking the learned policy to have poor performance [6]. Adaptive curriculum reinforcement learning (ACRL) addresses these issues by dynamically updating the training dataset [55]. ACRL generates and selects environments that yield the largest policy improvement. In our work, designing an effective environment generator is crucial. It should (1) generate realistic environments matching real-world distributions and (2) adequately challenge the current policy. Common approaches include using adjustable parametric terrain elevations [40], which offers control but may lack realism, and generative models [29], which excel in realism but may struggle with precise policy-tailored generation control. Meanwhile, those methods mostly focus on the bare terrain elevation, and robotic agile skills come from hand-crafted objects such as stairs and boxes [81]. Although radiance field rendering methods [47, 36] can bring the real-to-sim-to-real pipeline with powerful representation ability of digital twins, they also suffer from the training dataset diversity and scarcity, which limits the co-evolvement characteristics of the environment and policy.

C. Policy Distillation for Real-World Deployment

The policy learned in simulation can access both the noiseless state s and the global environment e. However, this *privileged* (ground-truth) information x is generally unavailable during real-world deployment due to robot sensors' measurement noise and limited field-of-view. Rather than employing model-free RL to train a deployment (student) policy within a simulation directly, most existing works prefer distilling this policy from the privileged one using imitation learning [40, 45]. Our approach aims to reduce the overly complicate demands of generating high-dimensional observations (e.g., noisy depth image simulation) and mitigate the deployment policy's risk of converging on local optima due to incomplete observations (e.g., historical encoding). Because the robustness of the deployment policy depends on both the performance of the privileged teacher policy and the diversity of its sensing observations derived from the training environments, it is important to have a diverse and realistic environment generator, which is the focus of this work.

IV. ADEPT: Adaptive Diffusion Environment for Policy Transfer

This section introduces the Adaptive Diffusion Environment for Policy Transfer, ADEPT, a novel ACRL generator in the zero-shot sim-to-real fashion that manipulates the DDPM process based on current policy performance and dataset diversity. We begin by interpolating between "easy" and "difficult" environments in the DDPM latent space to generate environments that optimize policy training. Next, we modulate the initial noise input based on the training dataset's variance to enrich environment diversity, fostering broader experiences and improving the policy's generalization across unseen environments. We use e, e_0, and e_k to denote the environment in the training dataset, the generated environment through DDPM, and the DDPM's latent variable at timestep k, respectively. All three variables are the same size, e.g., $e \in \mathbb{R}^{C\times W\times H}$. Since in DDPM, noises and latent variables are the same [25], we use them interchangeably.

A. Performance-Guided Generation via DDPM

We assume having access to a dataset Λ in the initial training phase, comprising N environments. The primary objective of

the adaptive environment generator is to dynamically create environments to be added to this dataset that optimally challenge the current policy. Ideally, these environments should push the boundaries of the policy's capabilities — being neither overwhelmingly difficult nor excessively simple for the policy to navigate. This approach ensures the training process is effective and efficient, promoting continuous learning and adaptation. We impose minimal constraints on the nature of initial environments, granting our method substantial flexibility in utilizing the available data. These environments can originate from various sources, such as elevation datasets, procedurally generated environments, or even those created by other generative models. We leverage the latent interpolation ability of DDPM to blend environments from the dataset to fulfill our objective. It adjusts the complexity of environments, simplifying those that are initially too challenging and adding complexity to simpler ones.

Latent Variable Synthesis for Controllable Generation. Once trained, DDPMs can control sample generation by manipulating intermediate latent variables. In our context, the goal is to *steer* the generated environments to maximize policy improvement after being trained on it. While there are numerous methods to guide the diffusion model [24, 65], we choose to optimize the starting noise to control the final target [5]. This approach is both simple and effective, as it eliminates the need for perturbations across all reverse diffusion steps, as required in classifier-free guidance [24], or fine-tuning of diffusion models [65]. Nevertheless, it still enhances the probability of sampling informative environments tailored to the current policy.

Consider a subset of environments $\bar{\Lambda} = (e^1, e^2, \ldots, e^n)$ from the dataset Λ, where the superscript means environment index rather than the diffusion step. To find an initial noise that generates an environment maximizing the policy improvement, we first generate intermediate latent variables (noises) for each training environment in $\bar{\Lambda}$ at a forward diffusion time step k, $e_k^i \sim q(e_k^i | e^i, k)$ for $i = 1, ..., n$. Assume that we have a weighting function $w(e, \pi)$ that evaluates the performance improvement after training on each environment e^i. We propose to find the optimized initial noise as a weighted interpolation of these latents, where the contribution of each latent e_k^i, $w(e^i, \pi)$, is given by the policy improvement in the original environment

$$e'_k = [\Sigma_{i=1}^n w(e^i, \pi) e_k^i] \ / \ [\Sigma_{m=1}^n w(e^m, \pi)]. \tag{2}$$

The fused latent variable e'_k is then processed through reverse diffusion, starting at time k to synthesize a new environment e'_0. The resulting environment blends the high-level characteristics captured by the latent features of original environments, proportionally influenced by their weights.

Weighting Function. The policy training requires dynamic weight assignment based on current policy performance. We define the following weighting function that penalizes environments that are too easy or too difficult for the policy:

$$\begin{aligned} w(e, \pi) &= \exp\{r(e, \pi)\}, \\ r(e, \pi) &= -(\mathfrak{s}(e, \pi) - \bar{\mathfrak{s}})^2 / \sigma^2. \end{aligned} \tag{3}$$

Specifically, it penalizes the deviation of *environment difficulty*, $\mathfrak{s}(e, \pi)$, experienced by the policy π from a desired difficulty level $\bar{\mathfrak{s}}$. This desired level indicates a environment difficulty that promotes the most significant improvement in the policy. The temperature parameter σ controls the sensitivity of the weighting function to deviations from this desired difficulty level. We use the navigation success rate [17] to represent $\mathfrak{s}(\cdot, \cdot)$. While alternatives like TD-error [32] or regret [52] exist, this metric has proven to be an effective and computationally efficient indicator for quantifying an environment's potential to enhance policy performance in navigation and locomotion tasks [45, 40]. We denote the procedure of optimizing the noise e'_k using Eq. (2) and generating the final optimized environment by reverse diffusion starting at e'_k as $e' = \texttt{Synthesize}(\bar{\Lambda}, \pi, k)$, where k is the starting time step of the reverse process. As discussed in the next section, a large k is crucial to maintaining diversity.

B. Diversifying Training Dataset via Modulating Initial Noise

The preceding section describes how policy performance guides DDPM in generating environments that challenge the current policy's capabilities. As training progresses, the pool of challenging environments diminishes, leading to a point where each environment no longer provides significant improvement for the policy. Simply fusing these less challenging environments does not create more complex scenarios. Without enhancing environment diversity, the potential for policy improvement plateaus. To overcome this, it is essential to shift the focus of environment generation towards increasing diversity. DDPM's reverse process generally starts from a predefined forward step, where the latent variable is usually pure Gaussian noise. However, it can also start from any forward step K with sampled noise as $e_K \sim q(e_K | e_0)$ [43]. To enrich our training dataset's diversity, we propose the following:

1) **Variability Assessment**: Compute the dataset's variability Λ_{var} by analyzing the variance of the first few principal components from a Principal Component Analysis (PCA) on each elevation map. This serves as an efficient proxy for variability.
2) **Forward Step Selection**: The forward step $k \propto \Lambda_{var}^{-1}$ is inversely proportional to the variance. We use a linear scheduler: $k = K(1 - \Lambda_{var})$, with K the maximum forward step and Λ_{var} normalized to $0 \sim 1$. This inverse relationship ensures greater diversity in generated environments.
3) **Environment Generation**: Using the selected forward step k, apply our proposed `Synthesize` to generate new environments, thus expanding variability for training environments.

C. ACRL with ADEPT

We present the final method pseudo-coded in Alg. 1 using the proposed ADEPT for training a privileged policy under

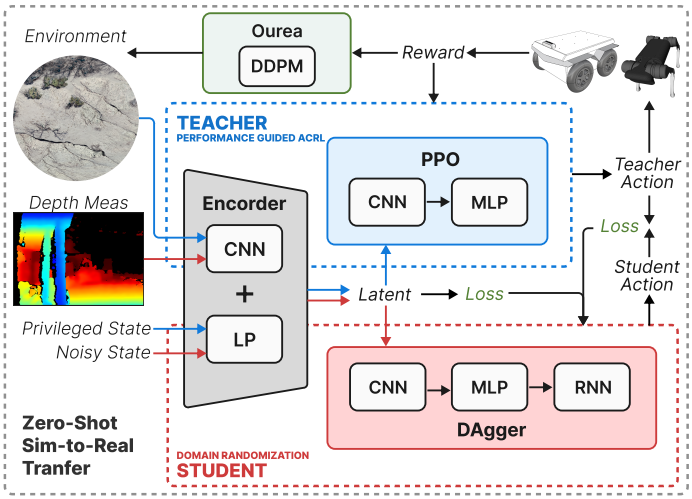

Fig. 1: Framework with our ADEPT and Policy Distillation. Model-free RL trains privileged policy ADEPT-generated environments. The privileged policy is then distilled into the deployment (Learner) policy using data aggregation. Iterative training and environment generation through ADEPT enhance the deployment policy's generalization.

Algorithm 1 ACRL with ADEPT

Input: Pretrained DDPM $\epsilon(\cdot, \cdot; \phi)$, an initial environment dataset Λ
Output: The optimized privileged policy π^*
Initialize: The privileged policy π

1: **while** π not converge **do**
2: $\quad e = \texttt{Selector}(\Lambda, \pi)$ ▷ Env. Selection
3: $\quad \pi \leftarrow \texttt{Optim}(\pi, e)$ ▷ Policy Update
4: $\quad k = K(1 - \Lambda_{var})$ ▷ Sec. IV-B
5: $\quad e'_0 = \texttt{Synthesize}(\Lambda, \pi, k)$ ▷ Sec. IV-A
6: $\quad \Lambda \leftarrow \Lambda \cup e'_0$ ▷ Update Dataset
7: **end while**

the adaptive curriculum reinforcement learning (ACRL). The algorithm iterates over policy optimization and guided environment generation, co-evolving the policy and environment dataset until convergence. The algorithm starts by selecting a training environment that provides the best training signal for the current policy, which can be done in various ways [6]. For example, one can compute scores for environments based on the weighting function in Eq. (3) and choose the one with the maximum weight. Instead of choosing deterministically, we sample the environments based on their corresponding weights. In practice, `Selector` bases its selections on the Upper Confidence Bound (UCB) algorithm, whose preference is defined as each environment's weight. `Optim` collects trajectories and performs one policy update in the selected environments. After the update, we evolve the current dataset by generating new ones, as shown in lines 4 - 6 of Alg. 1. Benefiting from the massively parallel simulator, we can run Alg. 1 in parallel across N environments, each with multiple robots. In parallel training, `Synthesize` begins by sampling $N \times n$ initial noises, where N is the number of new environments (equal to the number of parallel environments) and n is the sample size in Eq. (2). It then optimizes over these noises to generate N optimized noises. Finally, these optimized noises are passed to the DDPM to generate N environments. When the dataset grows large, it sub-samples environments from `Selector`'s complement, with success rates updated by the current policy.

D. ADEPT with Teacher-Student Distillation

We have introduced the adaptive diffusion environment for policy transfer, ADEPT, specifically designed to train a policy to generalize over environment geometries. However, as highlighted in Section III-C, real-world deployments face challenges beyond geometry, including noisy, partial observations and varying physical properties. To address these challenges, we distill the policy under teacher-student paradigm, within massively parallel simulating our proposed environment generator.

Environment Representation. An efficient and powerful representation for complicate off-road environments is necessary for DDPMs. It should (1) capture the complex details and filter out redundant information of real-world environments, and (2) balance the generation quality and computation (of training and inference) burden. Instead of signed distance function (SDF) or polygon mesh that have shown successes in indoor geometry generations [20, 21] but endure high computation costs because off-road environmental details need finer spatial resolutions, we propose a coarse-to-fine method that starts from environment generation via diffusion and then guides procedural generation to complete the details. First, our diffusion model encoding space is $e \in \mathbb{R}^{2\times W\times H}$ with two layers - `terrain elevation` and `surface canopy`. The first layer is the bare terrain elevation and the second layer describes the layout of wild plants. This representation is computationally fast and lightweight.

With the diffusion-generated environment e, the elevated terrain is extracted from `terrain elevation`. To reconstruct the wild plants from `surface canopy`, we firstly leverage the tree identifier [11] to map individual plants with each height and crown. As the segmented `surface canopy` shown in the middle of Fig. 2, we use the Convex hull to define the boundary for each plant and to guide the procedural generation to produce plant geometries. Specially, we sample points inside the convex hull and uses procedural growth to connect those points as branches and leaves to generate various plants, such as bushes and trees.

Teacher-Student Policy. To further address partial observations and varying physical property challenges above ADEPT-generated environments, we distill a teacher policy θ trained using PPO [60], which observes the privileged information x_t and noiseless state s_t at each timestamp t into a depth vision-based student policy $\hat{\theta}$ with noisy measurements $\hat{\theta}(\tilde{o}_t, \tilde{s}_t)$. The privileged information x_t includes the complete environment geometry, friction, restitution, gravity, and robot-environment contact forces. The state s_t, dependent on the embodiment, includes the robot motion information, which is usually esti-

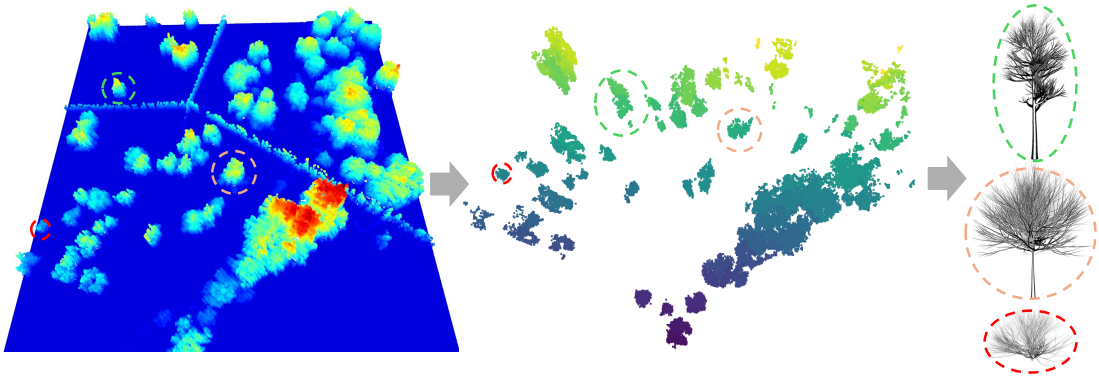

Fig. 2: The generation process of various plants from segmenting the surface canopy heights to procedurally generating plants within each extracted bounds. Those complex objects thus simulate to challenge the robot perception ability.

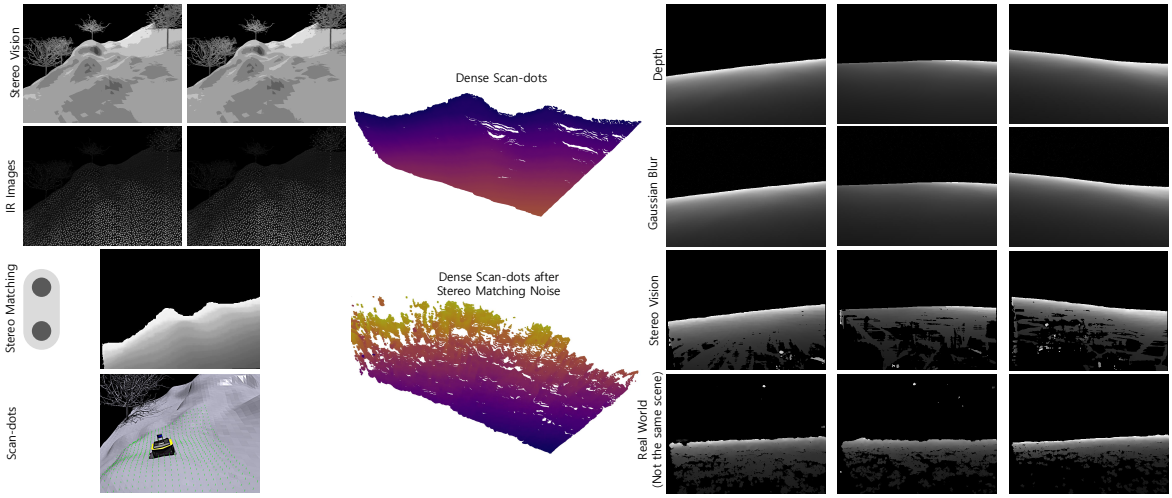

Fig. 3: Our proposed perception system mirrors the real active stereo-vision depth sensor pipeline to mitigate the sim-to-real gap. By projecting IR patterns onto rendered stereo images in the simulator and applying stereo matching to compute the disparity map, the resulting elevation noise is inherently tied to the stereo-vision depth noise rather than relying on hand-crafted values. The right panel illustrates examples of ground-truth depth, Gaussian blur (as a representative hand-crafted approach), noisy depth generated by stereo matching, and real-world depth noise patterns. Compared to the effects of Gaussian blur, our pipeline better reproduces realistic noise patterns.

mated with on-board sensors during deployment. Similarly, for each specified robot drive system, the applied action represents proportional-derivative (PD) targets $\alpha \cdot a_t^{\theta}$.

Student policy, $\hat{\theta}$, is trained via Dataset Aggregation (DAgger, [58, 1]) to match the teacher's actions with noisy and partially observable states. The policy has access to $(\tilde{s}_t, \tilde{o}_t)$, where $\tilde{s}_t$ is the noisy state and $\tilde{o}_t$ is the height scan [46]. We use height scans as they align with probabilistic elevation mapping [15], enabling multi-sensor fusion and supporting ground robot applications. Due to partial observability, the policy considers past information to decide the next action $a_t \sim \hat{\pi}(a_t|\boldsymbol{a}_t, \tilde{\mathbf{s}}_t, \tilde{\mathbf{o}}_t; \hat{\theta})$, where $\boldsymbol{a}_t$ and $\tilde{\mathbf{o}}_t$ are action and observation histories with the maximum history length H.

Domain Randomization. To enhance generalization, we integrate physics domain randomization and perception domain randomization. In the physics domain, an environment appears as geometry and is characterized by physics, including the friction, restitution, gravity, mass, external forces, and discrepancy in actuator set-points. These feature the robot-environment interaction and environmental properties.

In the perception domain, the state estimation uncertainty is modeled as independent Gaussian distributions, with covariance derived from the error upper bounds of modern SLAM systems [4, 8]. For exteroceptive perception, we propose simulating noise in two stages: first, by modeling depth measurement noise and then using it to generate noisy elevation maps. Instead of applying hand-crafted artifacts [1, 81], we simulate depth estimation errors based on active stereo sensor principles as shown in Fig. 3. Stereo-vision depth sensors provide crucial geometry without the sim-to-real challenges of RGB color alignment [76] and excel in accuracy and robustness due to infrared (IR) operation, simplifying simulation under randomized lighting compared to passive or RGB sensors. Using rendered stereo images, we introduce IR noise with the model [38]. Depth is estimated using four-path semi-global block matching (SGBM, [23]).

V. Sim-to-Deploy Experiments

We validate our method against competing approaches in both sim-to-sim and sim-to-real settings. Using wheeled and quadruped robot platforms, we assess its zero-shot transfer and generalization capabilities for challenging environments.

A. Algorithmic Performance Evaluation

This section validates the ADEPT framework on goal-oriented off-road navigation tasks, benchmarking its algorithmic performance against a popular method and assessing submodule contributions through ablation studies. These experiments serve as a prelude to the sim-to-deploy tests. This section evaluates whether the environment curriculum generated by ADEPT enhances the generalization capability of the trained privileged policy across unfamiliar environment geometries, on the wheeled ClearPath Jackal robot. We train in IsaacGym [42] and parallel 100 off-road environments, each with 100 robots. Simulations run on an NVIDIA RTX 4090 GPU.

We compare with the following baselines. Adaptive Procedural Generation (**APG**), a commonly used method, uses heuristically designed environment parameters [40]. Our implemented APG follows ADEPT, adapting the environment via the score function Eq. (3) and dynamically updating the dataset. First, to ablate our Adaptive curriculum, Diffusion Environment Policy Transfer (`A`**DEPT**) generates environment without curriculum. Procedural Generation (`A`**PG**) randomly samples parameters. To ablate our Diffusion Generator, Natural Adaptive Environment Policy Transfer (**N-A**`D`**EPT**) selects environments directly from E-3K. To ablate both, Natural Environment Policy Transfer (**N**-`AD`**EPT**) randomly samples from E-3K without curriculum. `Mono` font means the ablated parts. All methods use the same training and evaluation setup. After each training epoch, policies are tested in *held-out* evaluation environments with 60 000 start-goal pairs. Fig. 4 shows the normalized RL return, which is calculated by the actual return divided by the running bound. It reveals key takeaways of our ADEPT as following.

ADEPT generates realistic environments. The higher success rate of ADEPT than APG on the real-world replicated environments show the generation quality of ADEPT empowers robot navigation policy learning, compared to APG and N-AEPT. In the following sim-to-sim and sim-to-real experiments, we will demonstrate the smooth motion trained through ADEPT compared to the well-performing but unnatural policy from procedural generations.

ADEPT evolves environment difficulty. The RL return curve reflects the stable performance of ADEPT on evaluation environments, attributed to the evolving difficulty of training environments generated by ADEPT. As the policy encounters progressively harder environments, its performance initially dips but gradually stabilizes and converges. Although N-AEPT enjoys a large training dataset, it can hardly outperform ADEPT due to the lack of difficulty controllability.

ADEPT evolves environment diversity. ADEPT gains advantages over a fixed dataset such as N-AEPT because ADEPT can easily generate thousands of environments within tens of epochs. PG is limited by the parametric range and lacks efficient environment parameter control.

In summary, ADEPT excels at adapting environment difficulty and diversity based on evolving policy performance.

B. Sim-to-Deploy Experimental Setup

We benchmark on important metrics that include include the success rate, trajectory ratio, orientation vibration $|\omega|$, orientation jerk $|\frac{\partial^2 \omega}{\partial t^2}|$, and position jerk $|\frac{\partial a}{\partial t}|$, where ω and a denote the angular velocity and linear acceleration. These motion stability indicators are crucial in mitigating sudden pose changes. The trajectory ratio is the successful path length relative to straight-line distance and indicates navigator efficiency. All baselines use the elevation map [15] with depth camera and identify terrains as obstacles if the slope estimated from the elevation map exceeds $20°$.

We also compare with following state-of-the-art motion planners other than our ablations. **Falco** [78], a classic motion primitives planner, and Log-**MPPI** [48], a sampling-based model predictive controller, are recognized for the success rate and efficiency. They use the pointcloud and elevation map to weigh collision risk and orientation penalty. **TERP** [71], an RL policy trained in simulation, conditions on the elevation map, rewarding motion stability and penalizing steep slopes. **POVN**av [57] performs Pareto-optimal navigation by identifying sub-goals in segmented images [10], excelling in unstructured outdoor environments.

C. Simulation Experiment

We simulate wheeled robot, ClearPath Jackal, in ROS Gazebo on 30 diverse environments (E-30), equipped with a RealSense D435 camera (30 Hz). We add Gaussian noises to the ground-truth robot state (200 Hz), depth measurement, and vehicle control to introduce uncertainty whose parameters reflect the hardest curriculum during simulation training. The ROS message filter synchronized the odometry with depth measurement. 1000 start and goal pairs are sampled for each environment. We do not include ablations other than N-AEPT because of poor algorithmic performance. As results shown in Table I, our method outperforms the baselines. While all methods show improved performance due to the Husky's better navigability on uneven terrains, our method consistently outperformed baseline methods. The depth measurement noise poses a substantial challenge in accurately modeling obstacles and complex environments. Falco and MPPI often cause the robot to get stuck or topple over, and TERP often predicts erratic waypoints that either violate safety on elevation map or are overly conservative. Learning-based TERP and POVN lack generalizability, with their performance varying across different environments. This issue is mirrored in N-AEPT and APG, highlighting the success of adaptive curriculum and realistic environment generation properties of ADEPT.

D. Kilometer-Scale Field Trial.

In our real-world experiment, we implemented our student policy via zero-shot transfer on a Clearpath Jackal vehicle. The

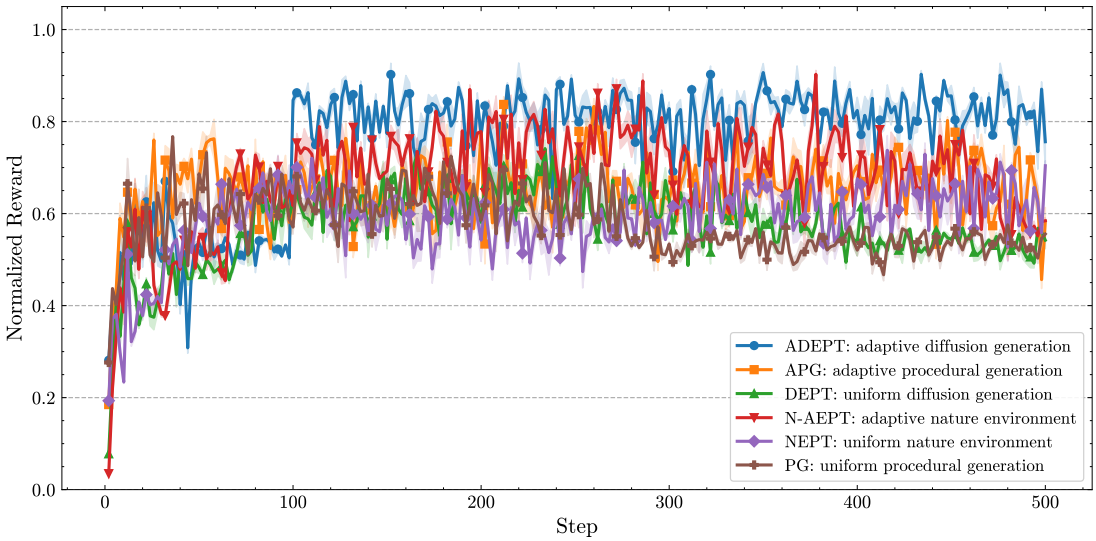

Fig. 4: The normalized return of our proposed ADEPT and the baseline methods on evaluation environments.

Jackal	Suc. Rate	Traj. Ratio	Orien. Vib. ($\mathrm{rad\,s^{-1}}$)	Orien. Jerk ($\mathrm{rad\,s^{-3}}$)	Pos. Jerk ($\mathrm{m\,s^{-3}}$)
Falco	0.26	2.76	0.71	275.56	47.95
MPPI	0.48	**1.21**	0.75	228.66	40.69
TERP	0.33	1.62	0.77	**210.05**	**37.08**
POVN	0.17	1.23	**0.68**	240.98	43.69
N-AEPT	**0.67**	**1.24**	1.08	323.37	57.63
APG	0.43	1.92	0.97	236.1	41.92
Ours	**0.87**	1.52	**0.65**	**193.45**	**34.93**

TABLE I: Statistical results for simulations are presented for ClearPath Jackal wheeled robot. The evaluation baselines involve Falco, MPPI, TERP, POVNav, and ablations with N-AEPT and APG. A total of 30 000 start-goal pairs are considered for each method. **Green** and **Bold** indicate the best and second-best.

Fig. 5: Three long-range trajectories of our method are presented, with each trajectory provided with only one distant goal. The start and goal points are represented by green and orange dots.

robot, running on NVIDIA Jetson Orin, was equipped with a Velodyne-16 LiDAR (10 Hz), a RealSense D435i camera (30 Hz), and a 3DM-GX5-25 IMU (200 Hz). Faster-LIO [4] provided LiDAR-Inertial odometry at 200 Hz.

Our experiment extends to evaluating the capability of our method in executing extended long-range trial in the field, a feature enabled by ADEPT to continuously evolve the environment. Note that during training we normalize all state variables except for the goal distance. We conducted 3 distinct field trials, each covering approximately 1.3 km. It is important to note that this experiment is not designed for direct comparative analysis with other methods, as they often rely on serialized waypoints (less than 10 meters each) for local navigation. The trajectories from these three trials are visualized on a satellite map in Fig. 5. In trial C, manual intervention was required for a sharp turn due to road crossing. The robot demonstrated its ability to adjust its heading for goal alignment, though orientation vibration levels were not minimal, indicating constant adjustments to navigate uneven terrains. It should be noted that our method cannot make a big turn in the trajectory without some waypoints (more than 100 meters each). The trials reveal that our method effectively extends its navigational capacity to long distances across uneven terrains.

VI. Conclusion

We propose ADEPT, an Adaptive Diffusion Environment Generator to create realistic and diverse environments based on evolving policy performance, enhancing RL policy's generalization and learning efficiency. To guide the diffusion model generation process, we propose optimizing the initial noises based on the potential improvements of the policy after being trained on the environment generated from this initial noise. Algorithmic performance shows ADEPT's performance in generating challenging but suitable environments over established methods such as commonly used procedural generation curriculum. Combined with domain randomization in a teacher-student framework, it trains a robust deployment policy for zero-shot transfer to new, unseen environments. Sim-to-deploy tests with an wheeled robot validate our approach against SOTA planning methods.

References

[1] Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Legged Locomotion in Challenging Terrains using Egocentric Vision. In *Proc. Conf. Robot Learn.*, number 2, pages 403–415, 2023.

[2] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

[3] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[4] Chunge Bai, Tao Xiao, Yajie Chen, Haoqian Wang, Fang Zhang, and Xiang Gao. Faster-LIO: Lightweight Tightly Coupled Lidar-Inertial Odometry Using Parallel Sparse Incremental Voxels. *IEEE Robot. & Automat. Letters*, 7 (2):4861–4868, 2022.

[5] Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-Flow: Differentiating through Flows for Controlled Generation. *arXiv preprint arXiv:2402.14017*, 2024.

[6] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Int. Conf. on Mach. Learn.*, pages 41–48, 2009.

[7] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

[8] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Trans. on Robot.*, 37(6):1874–1890, 2021.

[9] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by Cheating. In *Proc. of the Conf. on Robot Learning*, volume 100, pages 66–75. PMLR, 30 Oct–01 Nov 2020.

[10] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention Mask Transformer for Universal Image Segmentation. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1280–1289, 2022.

[11] Michele Dalponte and David A Coomes. Tree-centric mapping of forest carbon density from airborne laser scanning and hyperspectral data. *Methods in ecology and evolution*, 7(10):1236–1245, 2016.

[12] Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. *Advances in Neural Info. Processing Syst.*, 33:13049–13061, 2020.

[13] Alejandro Escontrela, Justin Kerr, Kyle Stachowicz, and Pieter Abbeel. Learning Robotic Locomotion Affordances and Photorealistic Simulators from Human-Captured Data. In *8th Annual Conference on Robot Learning*, 2024.

[14] Meng Fang, Tianyi Zhou, Yali Du, Lei Han, and Zhengyou Zhang. Curriculum-guided Hindsight Experience Replay. In *Advances in Neural Info. Processing Syst.*, volume 32. Curran Associates, Inc., 2019.

[15] Péter Fankhauser, Michael Bloesch, and Marco Hutter. Probabilistic Terrain Mapping for Mobile Robots with Uncertain Localization. *IEEE Robot. & Automat. Letters*, 3(4):3019–3026, 2018.

[16] Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic Goal Generation for Reinforcement Learning Agents. In *Proc. of Int. Conf. on Mach. Learn*, volume 80, pages 1515–1528. PMLR, 10–15 Jul 2018.

[17] Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic Goal Generation for Reinforcement Learning Agents. In *Int. Conf. on Mach. Learn.*, volume 80 of *Proceedings of Machine Learning Research*, pages 1515–1528. PMLR, 10–15 Jul 2018.

[18] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robot. & Automat. Magazine*, 4(1):23–33, 1997.

[19] Jonas Frey, Matias Mattamala, Nived Chebrolu, Cesar Cadena, Maurice Fallon, and Marco Hutter. Fast Traversability Estimation for Wild Visual Navigation. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.

[20] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021.

[21] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, 129:3313–3337, 2021.

[22] Xiefan Guo, Jinlin Liu, Miaomiao Cui, Jiankai Li, Hongyu Yang, and Di Huang. Initno: Boosting text-to-image diffusion models via initial noise optimization. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pages 9380–9389, 2024.

[23] Daniel Hernandez-Juarez, Alejandro Chacón, Antonio Espinosa, David Vázquez, Juan Carlos Moure, and Antonio M. López. Embedded Real-time Stereo Estimation via Semi-Global Matching on the GPU. In *International Conference on Computational Science 2016, ICCS 2016, 6-8 June 2016, San Diego, California, USA*, pages 143–153, 2016.

[24] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Info. Processing Syst.*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.

[26] David Hoeller, Nikita Rudin, Dhionis Sako, and Marco Hutter. ANYmal parkour: Learning agile navigation for quadrupedal robots. *Science Robotics*, 9(88):eadi7566, 2024.

[27] Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado van Hasselt, and David Silver. Distributed Prioritized Experience Replay. In *Int. Conf. on Learn. Representations*, 2018.

[28] Han Hu, Kaicheng Zhang, Aaron Hao Tan, Michael Ruan, Christopher Agia, and Goldie Nejat. A Sim-to-Real Pipeline for Deep Reinforcement Learning for Autonomous Robot Navigation in Cluttered Rough Terrain. *IEEE Robot. & Automat. Letters*, 6(4):6569–6576, 2021.

[29] Aryamaan Jain, Avinash Sharma, and Rajan. Adaptive & Multi-Resolution Procedural Infinite Terrain Generation with Diffusion Models and Perlin Noise. In *Proc. of the Thirteenth Indian Conference on Computer Vision, Graphics and Image Processing*, 2023. ISBN 9781450398220.

[30] Fabian Jenelten, Junzhe He, Farbod Farshidian, and Marco Hutter. DTC: Deep Tracking Control. *Science Robotics*, 9(86):eadh5401, 2024.

[31] Zhuozhu Jian, Zihong Lu, Xiao Zhou, Bin Lan, Anxing Xiao, Xueqian Wang, and Bin Liang. Putn: A plane-fitting based uneven terrain navigation framework. In *IEEE/RSJ Int. Conf. on Intel. Robots and Syst. (IROS)*, pages 7160–7166. IEEE, 2022.

[32] Minqi Jiang, Michael Dennis, Jack Parker-Holder, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Replay-guided adversarial environment design. *Advances in Neural Info. Processing Syst.*, 34:1884–1897, 2021.

[33] Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *Int. Conf. on Mach. Learn.*, pages 4940–4950. PMLR, 2021.

[34] Shirel Josef and Amir Degani. Deep Reinforcement Learning for Safe Local Planning of a Ground Vehicle in Unknown Rough Terrain. *IEEE Robot. & Automat. Letters*, 5(4):6748–6755, 2020.

[35] Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn, and Siyu Tang. Optimizing diffusion noise can serve as universal motion priors. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pages 1334–1345, 2024.

[36] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4), July 2023.

[37] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. 2021.

[38] Michael J. Landau, Benjamin Y. Choo, and Peter A. Beling. Simulating Kinect Infrared and Depth Images. *IEEE Transactions on Cybernetics*, 46(12):3018–3031, 2016.

[39] Hojin Lee, Junsung Kwon, and Cheolhyeon Kwon. Learning-based Uncertainty-aware Navigation in 3D Off-Road Terrains. In *Proc. Int. Conf. Robot. Automat.*, pages 10061–10068, 2023.

[40] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, 5 (47):eabc5986, 2020.

[41] Dexun Li, Wenjun Li, and Pradeep Varakantham. Diversity Induced Environment Design via Self-Play, 2023.

[42] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac Gym: High Performance GPU Based Physics Simulation For Robot Learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[43] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *Int. Conf. on Learn. Representation*, 2022.

[44] Xiangyun Meng, Nathan Hatch, Alexander Lambert, Anqi Li, Nolan Wagener, Matthew Schmittle, JoonHo Lee, Wentao Yuan, Zoey Chen, Samuel Deng, et al. TerrainNet: Visual Modeling of Complex Terrain for High-speed, Off-road Navigation. *Robotics: Science and Systems*, 2023.

[45] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(1):eabk2822, 2022.

[46] Takahiro Miki, Joonho Lee, Lorenz Wellhausen, and Marco Hutter. Learning to walk in confined spaces using 3D representation. *arXiv preprint arXiv:2403.00187*, 2024.

[47] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[48] Ihab S. Mohamed, Kai Yin, and Lantao Liu. Autonomous Navigation of AGVs in Unknown Cluttered Environments: Log-MPPI Control Strategy. *IEEE Robot. & Automat. Letters*, 7(4):10240–10247, 2022.

[49] Joseph Moyalan, Yongxin Chen, and Umesh Vaidya. Convex Approach to Data-Driven Off-Road Navigation via Linear Transfer Operators. *IEEE Robot. & Automat. Letters*, 8(6):3278–3285, 2023.

[50] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. Curriculum Learning for Reinforcement Learning Domains: A Framework and Survey. *J. Mach. Learn. Res.*, 21(1), 2020. ISSN 1532-4435.

[51] Yunpeng Pan, Ching-An Cheng, Kamil Saigol, Keuntaek Lee, Xinyan Yan, Evangelos A Theodorou, and Byron Boots. Imitation learning for agile autonomous driving. *The International Journal of Robotics Research*, 39(2-3):

286–302, 2020.

[52] Jack Parker-Holder, Minqi Jiang, Michael Dennis, Mikayel Samvelyan, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Evolving curricula with regret-based environment design. In *Int. Conf. on Mach. Learn.*, pages 17473–17498. PMLR, 2022.

[53] Manthan Patel, Jonas Frey, Deegan Atha, Patrick Spieler, Marco Hutter, and Shehryar Khattak. Roadrunner m&m–learning multi-range multi-resolution traversability maps for autonomous off-road navigation. *arXiv preprint arXiv:2409.10940*, 2024.

[54] Utsav Patel, Nithish K Sanjeev Kumar, Adarsh Jagan Sathyamoorthy, and Dinesh Manocha. DWA-RL: Dynamically Feasible Deep Reinforcement Learning Policy for Robot Navigation among Mobile Obstacles. In *IEEE Int. Conf. on Robot. and Automat. (ICRA)*, pages 6057–6063, 2021.

[55] Rémy Portelas, Cédric Colas, Lilian Weng, Katja Hofmann, and Pierre-Yves Oudeyer. Automatic curriculum learning for deep RL: A short survey. *arXiv preprint arXiv:2003.04664*, 2020.

[56] Rémy Portelas, Cédric Colas, Lilian Weng, Katja Hofmann, and Pierre-Yves Oudeyer. Automatic curriculum learning for deep rl: A short survey. *arXiv preprint arXiv:2003.04664*, 2020.

[57] Durgakant Pushp, Zheng Chen, Chaomin Luo, Jason M. Gregory, and Lantao Liu. POVNav: A Pareto-Optimal Mapless Visual Navigator, 2023.

[58] Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *Proc. of Machine Learn. Research*, volume 15, pages 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

[59] Christoph Rösmann, Frank Hoffmann, and Torsten Bertram. Timed-Elastic-Bands for time-optimal point-to-point nonlinear model predictive control. In *European Control Conf.*, pages 3352–3357, 2015.

[60] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, 2017.

[61] Sriram Siva, Maggie Wigness, John Rogers, and Hao Zhang. Enhancing Consistent Ground Maneuverability by Robot Adaptation to Complex Off-Road Terrains. In *Conf. on Robot Learn.*, 2021.

[62] Kyle Stachowicz, Lydia Ignatova, and Sergey Levine. Lifelong Autonomous Improvement of Navigation Foundation Models in the Wild. In *8th Annual Conference on Robot Learning*, 2024.

[63] Samuel Triest, Mateo Guaman Castro, Parv Maheshwari, Matthew Sivaprakasam, Wenshan Wang, and Sebastian Scherer. Learning Risk-Aware Costmaps via Inverse Reinforcement Learning for Off-Road Navigation. In *Int. Conf. on Robot. and Automat.*, pages 924–930, 2023.

[64] Samuel Triest, Matthew Sivaprakasam, Shubhra Aich, David Fan, Wenshan Wang, and Sebastian Scherer. Velociraptor: Leveraging Visual Foundation Models for Label-Free, Risk-Aware Off-Road Navigation. In *8th Annual Conference on Robot Learning*, 2024.

[65] Masatoshi Uehara, Yulai Zhao, Tommaso Biancalani, and Sergey Levine. Understanding Reinforcement Learning-Based Fine-Tuning of Diffusion Models: A Tutorial and Review. *arXiv preprint arXiv:2407.13734*, 2024.

[66] Siddarth Venkatraman, Moksh Jain, Luca Scimeca, Minsu Kim, Marcin Sendera, Mohsin Hasan, Luke Rowe, Sarthak Mittal, Pablo Lemos, Emmanuel Bengio, et al. Amortizing intractable inference in diffusion models for vision, language, and control. *arXiv preprint arXiv:2405.20971*, 2024.

[67] Jingping Wang, Long Xu, Haoran Fu, Zehui Meng, Chao Xu, Yanjun Cao, Ximin Lyu, and Fei Gao. Towards Efficient Trajectory Generation for Ground Robots beyond 2D Environment. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 7858–7864, 2023.

[68] Linji Wang, Zifan Xu, Peter Stone, and Xuesu Xiao. Grounded Curriculum Learning. *arXiv preprint arXiv:2409.19816*, 2024.

[69] Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O Stanley. Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions. *arXiv preprint arXiv:1901.01753*, 2019.

[70] Rui Wang, Joel Lehman, Aditya Rawal, Jiale Zhi, Yulun Li, Jeff Clune, and Kenneth O. Stanley. Enhanced POET: Open-Ended Reinforcement Learning through Unbounded Invention of Learning Challenges and their Solutions. In *Int. Conf. on Mach. Learn.*, 2020.

[71] Kasun Weerakoon, Adarsh Jagan Sathyamoorthy, Utsav Patel, and Dinesh Manocha. TERP: Reliable Planning in Uneven Outdoor Environments using Deep Reinforcement Learning. In *Proc. Int. Conf. Robot. Automat.*, pages 9447–9453, 2022.

[72] Grady Williams, Paul Drews, Brian Goldfain, James M. Rehg, and Evangelos A. Theodorou. Aggressive driving with model predictive path integral control. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 1433–1440, 2016.

[73] Xuesu Xiao, Joydeep Biswas, and Peter Stone. Learning Inverse Kinodynamics for Accurate High-Speed Off-Road Navigation on Unstructured Terrain. *IEEE Robot. & Automat. Letters*, 6(3):6054–6060, 2021.

[74] Junhong Xu, Kai Yin, Zheng Chen, Jason M Gregory, Ethan A Stump, and Lantao Liu. Kernel-based diffusion approximated Markov decision processes for autonomous navigation and control on unstructured terrains. *The International Journal of Robotics Research*, page 02783649231225977, 2024.

[75] Long Xu, Kaixin Chai, Zhichao Han, Hong Liu, Chao Xu, Yanjun Cao, and Fei Gao. An Efficient Trajectory Planner for Car-Like Robots on Uneven Terrain. In *IEEE/RSJ Int. Conf. on Intel. Robots and Syst. (IROS)*, pages 2853–2860. IEEE, 2023.

[76] Alan Yu, Ge Yang, Ran Choi, Yajvan Ravan, John

Leonard, and Phillip Isola. Learning Visual Parkour from Generated Images. In *8th Annual Conference on Robot Learning*, 2024.

[77] Youwei Yu, Junhong Xu, and Lantao Liu. Adaptive Diffusion Terrain Generator for Autonomous Uneven Terrain Navigation. In *8th Annual Conference on Robot Learning*, 2024.

[78] Ji Zhang, Chen Hu, Rushat Gupta Chadha, and Sanjiv Singh. Falco: Fast likelihood-based collision avoidance with extension to human-guided navigation. *Journal of Field Robot.*, 37:1300 – 1313, 2020.

[79] Xiaojing Zhang, Alexander Liniger, and Francesco Borrelli. Optimization-Based Collision Avoidance. *IEEE Trans. on Control Sys. Tech.*, 29(3):972–983, 2021.

[80] Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. Guided conditional diffusion for controllable traffic simulation. In *IEEE Int. Conf. on Robot. and Automat. (ICRA)*, pages 3560–3566. IEEE, 2023.

[81] Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher G Atkeson, Sören Schwertfeger, Chelsea Finn, and Hang Zhao. Robot Parkour Learning. In *Conf. on Robot Learn.*, 2023.