VANDERER: Map-Free Exploration using Future-Aware and Visual-Curiosity-Guided Diffusion Policy

Abstract— **Mobile agents require efficient exploration strategies to map unseen environments and autonomously plan tasks. Traditional methods rely on generating occupancy maps and optimizing the sequence in which unexplored regions are visited. However, in sensor-constrained settings, such as those limited to monocular cameras, generating accurate occupancy maps is challenging. To address this, we propose VANDERER, an exploration framework that leverages a Visual Curiosity Module (VCM) to guide pre-trained diffusion policies using only monocular image data. This curiosity module predicts the outcomes of proposed actions via a navigation world model and evaluates them through a curiosity cost. The cost then guides the diffusion process toward generating actions that maximize exploration. Evaluated across diverse simulated environments, VANDERER consistently outperforms established baselines, exploring an average of 13.4% more area than NoMaD [1]. Our results reveal a direct correlation between visual and geometric curiosity in outdoor environments, demonstrating that VANDERER can effectively leverage this relationship for efficient exploration using sensor-constrained agents.**

I. INTRODUCTION

Autonomous exploration is a long-standing problem in robotics, with critical applications in search-and-rescue, environmental monitoring, and infrastructure inspection. Traditional frameworks rely on constructing global occupancy maps, necessitating precise localization and high-fidelity sensors (GPS, IMUs, depth cameras, or LiDAR) for accurate positioning and spatial mapping. This imposes significant hardware constraints on the physical agent. Furthermore, while frontier-based methods [2] are effective in structured indoor settings, outdoor environments introduce unique complications. For instance, geometric gaps between buildings may be flagged as frontiers even when they are physically impassable. Moreover, outdoor exploration is often governed by semantic or legal constraints, such as traffic regulations, which restrict movement to specific zones regardless of geometric accessibility. To address such limitations, this work tackles the problem of outdoor exploration while foregoing the need for expensive LiDARs, IMUs, and GPS systems.

Recent works in exploration and navigation have increasingly centered on learning-based approaches. A significant portion of such research relies on Reinforcement Learning (RL) to train task-specific policies [3]. However, RL-based methods typically require extensive training within individual environments and demand large amounts of high-quality data [4]. Consequently, diffusion policies [5] have emerged as a promising alternative, often reducing the training time required to adapt policies to new environments. Recent results demonstrate their success in tasks such as goal-directed navigation [1], exploration [6], and collision-free planning [7]. Furthermore, diffusion policies have shown potential for generalizing across multiple tasks through intelligent guidance and masking strategies [1].

Environment exploration, however, presents a unique challenge. Unlike goal-directed navigation, where training trajectories are abundant, expert data for exploration is limited. In fact, exploration is not a strictly defined objective with a single optimal behavior. Rather, it is a broad goal that admits many valid trajectories. This raises a fundamental question: how can large-capacity models be trained to behave exploratively without relying on explicit expert exploration data? Additionally, how can existing datasets or scalable data generation be used to support such training?

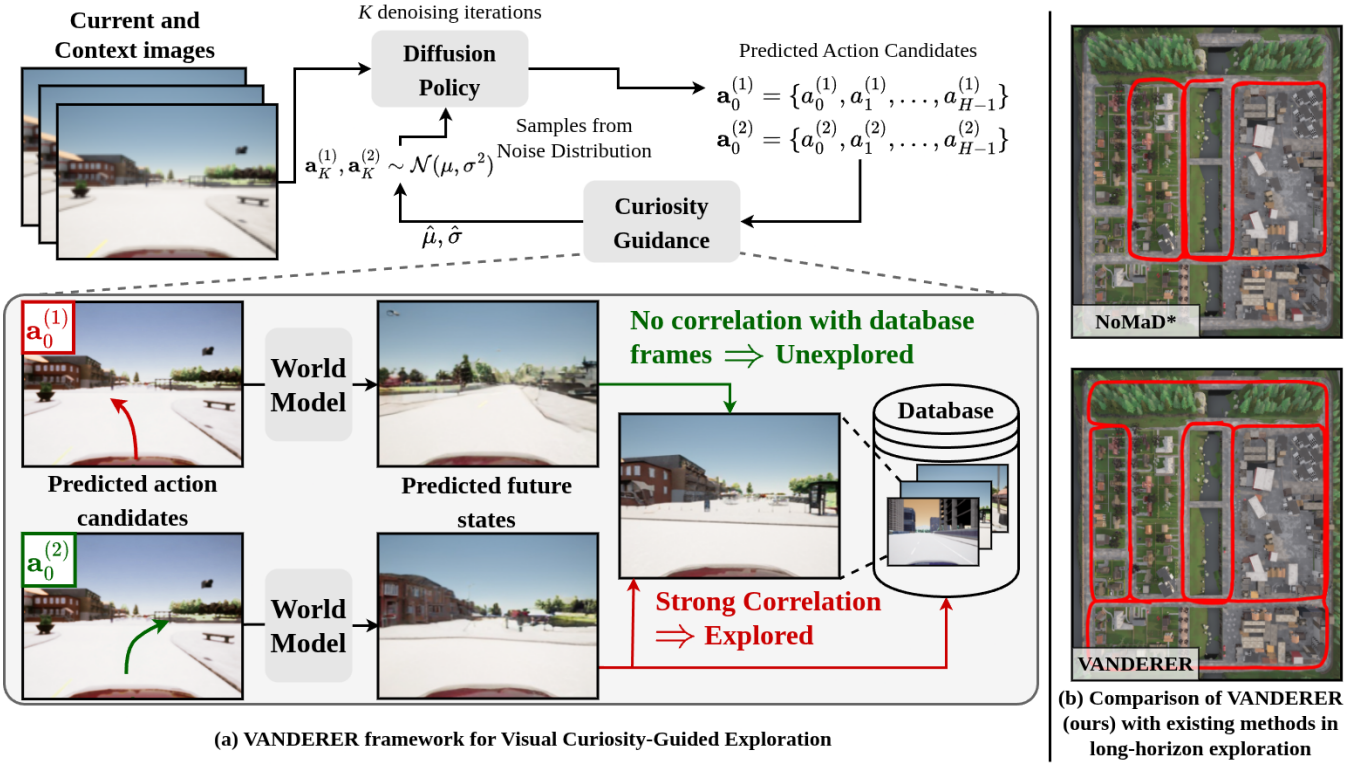
To address these challenges, we propose VANDERER (Fig. 1), a framework designed for efficient outdoor exploration that foregoes the overhead of complex mapping. Our approach leverages a diffusion policy to generate feasible candidate low-level actions directly from RGB observations. Central to our method is the Visual Curiosity Module (VCM), which utilizes a navigation world model [8] to predict future states resulting from candidate actions. The VCM quantifies the "exploratory value" of these predicted states by comparing them against a database of prior observations via a curiosity-based cost function. By assigning lower costs to novel states, the VCM acts as diffusion guidance, steering the sampling process toward exploratory trajectory generation during inference. Crucially, VANDERER eliminates the need for specialized exploration data. We evaluate our method through comparisons with competitive baselines and ablative studies. Throughout the evaluation, we treat all the baselines the exact same way, maintaining consistency on the assumptions, data, and finetuning of their respective models. In summary, our key contributions are:

- A novel end-to-end framework for map-free exploration using a visual-curiosity-guided diffusion policy.
- A visual curiosity module that predicts the consequences of actions and guides the diffusion policy to generate outputs with greater exploratory potential.
- An inference-stage optimization strategy that adapts pre-trained diffusion policies for efficient exploration without the need for additional training.
- Extensive experimentation showing VANDERER's superior exploration efficiency over existing methods.

II. RELATED WORKS

A. Environment Exploration

Environment exploration methods can be broadly divided into two paradigms: traditional mapping-based and learning-



(a) VANDERER framework for Visual Curiosity-Guided Exploration

(b) Comparison of VANDERER (ours) with existing methods in long-horizon exploration

Fig. 1: VANDERER framework for guiding diffusion policies towards efficient autonomous exploration. (a) A simplified example of action predictions generated from two noise latents sampled from a time-invariant Gaussian distribution through a pretrained diffusion policy. The curiosity guidance mechanism predicts the resulting future states and estimates their novelty against a database of prior observations. These novelty scores are then used to update the diffusion policy's noise distribution, guiding the agent toward unexplored areas. (b) VANDERER enables the agent to explore larger environments compared to existing state-of-the-art methods.

based approaches. *Mapping-based exploration* focuses on expanding observed space by identifying navigable points. Frontier search methods [2], [9] target occupancy grid boundaries, while sampling-based approaches [10], [11] explore unoccluded areas via tree or graph structures [12], [13]. Hybrid frameworks [14] attempt to balance these by combining frontier guidance with sampling-based local exploration. Crucially, all these methods rely on accurate map building using multiple sensors, a hardware constraint our monocular RGB approach explicitly removes.

A prominent class of *learning-based exploration* approaches utilize RL frameworks to train policies through various intrinsically motivated reward structures, such as future-state prediction error in a learned latent space [15]. Curiosity can also be measured using network distillation that compares the output of a trained predictor with that of an untrained one, using the error as a signal for curiosity [16]. A more recent approach uses the error between the predictions of an ensemble of single-step dynamic models as curiosity reward [17]. In contrast, several deep reinforcement learning methods [18] use extrinsic reward obtained from additional sensors such as depth cameras and LiDARs to abstract information like explorative frontiers and occupancy maps [19]–[21]. While these strategies are effective, the requirement for environment-specific training in simulation often limits their direct applicability to real-world exploration.

B. Diffusion Policies

Analogous to map-based extrinsic rewards in RL, diffusion policies can learn exploration by conditioning on map representations [6], [22]. More specifically, recent works have utilized raw RGB frames from real-world trajectories to learn goal-based navigation [23], [24]. Diffusion policies have further enabled unified frameworks that learn both navigation and exploration within a single model [1], with subsequent improvements incorporating learned metric scaling [25] and conditional flow matching [26]. While these RGB-only methods perform well for goal-conditioned tasks, they rely primarily on policy variance for discovery, lacking the global information necessary for long-horizon exploration.

Another advantage of diffusion policies is their capacity for objective-specific guidance using pre-trained models. This can be achieved through trained classifiers that steer the denoising loop [27] or via classifier-free guidance, which uses a weighted combination of class-dependent and class-agnostic generations [28]. Recent research has also focused on optimizing the initial noise distribution for more targeted generation [29]–[31]. In goal-based navigation, guidance techniques have already been applied to collision avoidance and safety [7], [32]. Our work builds on these methods, proposing a curiosity-based guidance mechanism that optimizes diffusion outputs for long-range exploration using only a monocular RGB stream.

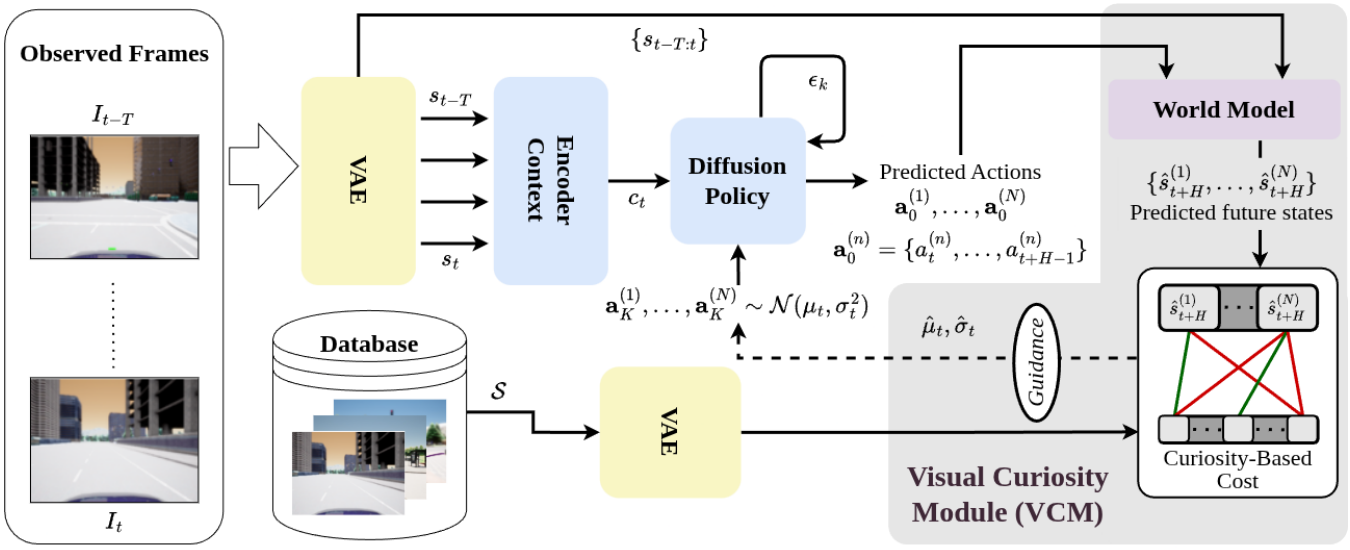


Fig. 2: The current (I_t) and T prior frames ($I_{t-T:t-1}$) are separately encoded using a Variational Autoencoder (VAE). A context encoder generates a conditioning vector from the states for policy generation. The world model in the VCM autoregressively predicts sequences of future states using the observed frames and policy-generated action sequences. The curiosity-based cost function matches each predicted final state with the previously observed states in the database. The most novel candidate is used to update the diffusion noise distribution.

III. METHODOLOGY

Problem Statement. We consider an agent equipped with a monocular camera traversing an unmapped, city-scale outdoor environment. At each discrete timestep t, the agent receives an RGB observation $I_t \in \mathbb{R}^{\mathcal{H}\times\mathcal{W}\times 3}$, ($\mathcal{H}$: height; $\mathcal{W}$: width of the image) which serves as the sole input for navigation, exactly like NoMaD [1]. The objective of VANDERER is to generate navigation commands that maximize exploration efficiency, which is defined as the ratio of total explored area to cumulative distance traveled. These commands are represented as waypoints in the 2D top-down projection of the agent's local coordinate frame. We quantify the explored area by discretizing the environment into a top-down grid and calculating the total area of all cells traversed by the agent. Finally, the generated waypoints are executed on the agent using a low-level controller.

Method Overview. Existing approaches (e.g., [1]) typically focus on goal-conditioned navigation and use future frames as goals during training. However, the resulting paths lack global context and are rarely optimized for exploration. Further, scaling 2D occupancy grids, capable of generating exploratory trajectories, to large outdoor environments is difficult due to complex global mapping requirements. To address these limitations, we employ a "generate-then-optimize" framework. Our diffusion policy is trained to imitate trajectory data to predict feasible and safe action sequences. During inference, we guide the model toward exploration-friendly behaviors by optimizing the initial noise latent distribution within the reverse diffusion process.

At the core of VANDERER is the VCM (Sec. III-A), which guides a Diffusion Policy (Sec. III-B) to generate actions tailored for environmental exploration (Fig. 2). VANDERER first maps the current observation I_t and T historical frames to latent states $\{s_{t-T}, \ldots, s_t\}$ using patch-level features from a frozen VAE [8], [33]. These states are encoded into a conditioning vector c_t, allowing the diffusion policy to iterate from a large denoising timestep (K) to 0, eventually producing an action sequence $\{a_0, \ldots, a_{H-1}\}$ representing relative (x, y) coordinate changes. Sampling multiple noise latents, N such action candidates are generated. In notation, a variable $x_t^{(n)}$ denotes the n-th sample of the variable x at time t. Further, bold lettered actions $\mathbf{a}_k^{(n)}$ denotes the n-th action sequence sample generated at the k-th denoising timestep by the diffusion policy. Finally, the VCM processes these actions and states to predict future observations, which inform a latent guidance loss that optimizes the diffusion process for maximum environmental exploration.

A. Visual Curiosity Guidance

The VCM predicts the consequences of diffusion-generated actions to steer the agent toward exploration. This mechanism comprises three core components: future state prediction via a world model, a curiosity-based cost function, and the application of this loss to provide diffusion guidance.

Predicting Future States. The action sequence generated by the diffusion policy, combined with context and current latent states, is fed into a world model to predict (*imagine*) future latent states $\{\hat{s}_{t+1}, \ldots, \hat{s}_{t+H}\}$. We employ the Navigation World Model (NWM) framework [8], which utilizes patch-level VAE features as states and defines the action vector as the relative change (Δ) in x-coordinate, y-coordinate, and yaw. With x and y displacement values from a_i, we estimate the change in yaw as $\arctan(\Delta y/\Delta x)$.

Curiosity-Based Cost Function. Given the predicted future states, we want to quantify their exploratory value. In the absence of global maps or odometry, we assess novelty by

Algorithm 1 Curiosity Loss Computation

Input: Predicted future states $\{\hat{s}_{t+H}^{(n)}\}_{n=1}^{N}$, Database $\mathcal{S}$
Output: Curiosity costs
costs $\leftarrow [\,]$
for each predicted future state $\hat{s}_{t+H}^{(n)}$ **do**
  distances $\leftarrow [\,]$
  for each visited state $s \in \mathcal{S}$ **do**
    dists $\leftarrow$ fast_reciprocal_nearest_neighbors$(\hat{s}_{t+H}^{(n)}, s)$
    $d \leftarrow \text{mean}(\text{topk}(\textit{dists}, 250))$
    append d to *distances*
  end for
  $\mathcal{L}_c^{(n)} \leftarrow \min(\textit{distances})$
  append $\mathcal{L}_c^{(n)}$ to *costs*
end for
return *costs*

```
def vanderer(current_img, context_imgs, database):
    # noise_distribution: init as standard normal

    # Convert to latent state representation
    latent_states = VAE(current_img, context_imgs)

    # 1. Sample N noise latents
    noise_latents = noise_distribution.sample(N)

    # 2. Diffusion Policy
    c_t = context_transformer(latent_states)
    actions = noise_prediction(c_t, noise_latents)

    # 3. Curiosity Guidance through VCM
    mean_updated, var_updated = VCM(actions, latent_states,
                        database, noise_latents)

    # 3. CEM Update & Resample
    noise_distribution = Normal(mean_updated, var_updated)

    action_updated = policy(noise_distribution.sample(1))

    return action_updated
```

Fig. 3: Pseudocode of a single-step action guidance using VCM. The guidance algorithm initiates once the action candidates exhibit sufficient variance, at which point the noise distribution is updated based on VCM scores. Finally, the optimized action is reconstructed by the diffusion policy model using this refined noise distribution.

comparing the final predicted state $\hat{s}_{t+H}$ against the database of previously observed states $\mathcal{S} = \{s_0, \ldots, s_t\}$. Since these latent states correspond to visual observations, we hypothesize that in environments with sufficient visual variance, maximizing visual novelty effectively maximizes environmental coverage. To compute the similarity $\mathcal{D}$ between $\hat{s}_{t+H}$ and a reference state $s_j \in \mathcal{S}$, we utilize MASt3R's [34] fast reciprocal matching on patch-level features (Algorithm. 1). Specifically, an iterative reciprocal nearest neighbor search establishes a one-to-one mapping between patches in $\hat{s}_{t+H}$ and their closest counterparts in s_j. The similarity $\mathcal{D}$ is defined as the mean Euclidean distance between the *top 250* most distant corresponding patch pairs. The final curiosity cost $\mathcal{L}_c$ is determined by the minimum distance across all states in $\mathcal{S}$, identifying the observation most similar to the predicted future state:

$$\mathcal{L}_c(\hat{s}_{t+H}, \mathcal{S}) = \min_{s_j \in \mathcal{S}} \mathcal{D}(\hat{s}_{t+H}, s_j). \tag{1}$$

This patch-matching approach is more robust than pixel-wise or index-based comparisons, where minor camera shifts can cause significant errors if indices no longer align with the same visual features. Additionally, visual similarity metrics typically saturate beyond a certain threshold, failing to reflect true spatial separation between distant, distinct locations. To improve efficiency and reduce computational load, we compare the predicted state $\hat{s}_{t+H}$ against a temporally subsampled subset of $\mathcal{S}$, thereby filtering out the redundant information found in adjacent frames.

Diffusion Guidance. The curiosity cost $\mathcal{L}_c$ guides the diffusion process toward actions that prioritize exploration (Fig.3). Specifically, $\mathcal{L}_c$ is used to refine the mean and variance of the diffusion policy's noise input, as different noise latents yield distinct, feasible action sequences. Although the noise could be updated via various optimization strategies, we find that sampling-based methods like the Cross-Entropy Method (CEM) yield superior results, consistent with recent world model literature [8], [35]. The policy initially draws N noise samples from a standard Gaussian distribution ($\mathcal{N}(\mu_0 = 0, \sigma = \mathbb{I}_{2H})$), generating N candidate action sequences that are subsequently evaluated via $\mathcal{L}_c$. From these, the top-r performing candidates are selected to empirically calculate the updated mean μ_1 of the noise distribution. This process continues for L iterations. Finally, the noise sampled from this updated distribution is fed to the policy to generate the optimal action sequence. To maintain generation stability, the noise distribution is reset to a standard Gaussian every P execution steps. Notably, guidance is only applied when there is sufficient variance among the generated action candidates to ensure effective steering.

B. Diffusion Policy

To employ the diffusion policy, all frames are first processed by a VAE encoder into 3D feature tokens, which are then positionally encoded to preserve spatial and temporal properties. These tokens are processed by a context transformer, comprising six standard transformer blocks with 8 attention heads in a 256-dimensional space, before being pooled into a 1D vector to serve as conditioning for the diffusion model.

This model utilizes a UNet-style architecture. During training, following the DDPM framework, Gaussian noise is added to ground-truth action sequences according to Eq. 4 using a variance schedule for K diffusion timesteps, defined by parameters $(\beta_1, \ldots, \beta_K)$ and the following constants:

$$\alpha_k = 1 - \beta_k, \quad \bar{\alpha}_k = \prod_{i=1}^{k} \alpha_i, \quad \tilde{\beta}_k = \frac{1 - \bar{\alpha}_{k-1}}{1 - \bar{\alpha}_k} \beta_k \tag{2}$$

$$\epsilon \sim \mathcal{N}(0, \mathbb{I}) \quad k \in \{1, K\}. \tag{3}$$

Following Eq.4, noisy inputs $(\mathbf{a}_k)$ are generated from ground truth actions. The noise prediction model $(\epsilon_\theta(\mathbf{a}_k, k))$ is

TABLE I: Mean performance metrics across five environments (Towns 1-5). Area: Total Area (m^2), APL: Area per Path Length (m), Avg PF: Average Policy Failure across towns.

		Town 1		Town 2		Town 3		Town 4		Town 5	
Method	Avg PF	Area	APL	Area	APL	Area	APL	Area	A/PL	Area	APL
NoMaD [1]	0.278%	11589	2.95	4368	2.23	10827	3.42	15157	3.87	15088	3.84
DP-RND [16]	0.084%	10865	2.76	3205	1.63	10357	2.63	15301	3.90	13525	3.45
NoMaD* [1]	**0.042%**	8923	2.26	4229	2.14	12848	3.28	16459	4.19	14491	3.68
VANDERER	0.046%	**12299**	**3.12**	**5488**	**2.79**	**14267**	**3.65**	**17125**	**4.36**	**15525**	**3.93**

optimized to estimate this noise (refer to Eq. 5), given the noisy input:

$$\mathbf{a}_k = \sqrt{\bar{\alpha}_k}\mathbf{a}_0 + \sqrt{1-\bar{\alpha}_k}\epsilon \tag{4}$$

$$\mathcal{L}_{diff} = \mathbb{E}_{k,\mathbf{a}_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(\mathbf{a}_k, k)\|^2\right]. \tag{5}$$

During inference, the action prediction at any rollout timestep t begins by initializing the noisy input ($\mathbf{a}_K \sim \mathcal{N}(0, \mathbb{I})$), which is iteratively refined to obtain the denoised action sequence $\mathbf{a}_0$. At denoising timestep k, the estimated noise $\epsilon_\theta(\mathbf{a}_k, k)$ is used to predict the clean action ($\hat{\mathbf{a}}_0$), which in turn is used to compute $\mathbf{a}_{k-1}$. This process continues until $k = 1$ for $K = 10$ timesteps, to derive the final denoised output using the following equations:

$$\hat{\mathbf{a}}_0 = \frac{\mathbf{a}_k - \sqrt{1-\bar{\alpha}_k}\epsilon_\theta(\mathbf{a}_k, k)}{\sqrt{\bar{\alpha}_k}} \tag{6}$$

$$\mathbf{a}_{k-1} \sim \mathcal{N}\left(\tilde{\mu}_k(\mathbf{a}_k, \hat{\mathbf{a}}_0), \tilde{\beta}_k\mathbb{I}\right) \quad \forall k \in \{K, \ldots, 1\} \tag{7}$$

$$\tilde{\mu}_k(\mathbf{a}_k, \hat{\mathbf{a}}_0) = \frac{\sqrt{\bar{\alpha}_{k-1}}\beta_k}{1-\bar{\alpha}_k}\hat{\mathbf{a}}_0 + \frac{\sqrt{\alpha_k}(1-\bar{\alpha}_{k-1})}{1-\bar{\alpha}_k}\mathbf{a}_k. \tag{8}$$

IV. EXPERIMENTS

Our experiments are designed to address the following key questions: (a) How effectively does VANDERER navigate and explore large-scale outdoor environments? (b) What is the impact of curiosity-based diffusion guidance on overall performance? (c) How does the explored area scale with respect to experiment duration? (d) Which components of VANDERER are most critical to its success?

A. Experiment Settings

Dataset. We evaluate VANDERER using the CARLA [36] simulator, which provides a robust testing suite of multiple outdoor environments featuring varied scales and diverse visual features. Our experiments utilize five "town" environments, where the agent is a vehicle equipped with an ego monocular camera. To simplify the environments, we removed lamp posts from the scenes. The training data consists of collision-free trajectories generated via CARLA's *BasicAgent* between random waypoints within a set distance threshold. We evaluate each environment across three random seeds to ensure consistency.

Training Settings. We trained the Diffusion Policy (Sec. III-B) using a single NVIDIA L40s GPU with a batch size of 256 for 100 epochs. To optimize performance, we further fine-tuned this base model for 20 epochs on each individual town, using these specialized models for their respective evaluations. All the other baselines are also treated the exact same way to ensure a fair comparison. Simultaneously, we fine-tuned the "small" version of the Navigation World Model [8] utilized in the VCM (Sec. III-A). Rather than fine-tuning separate models for each town, we fine-tuned a single model from its original checkpoints on the collective five-town dataset for 55 epochs using the same compute resources. During evaluation, proposed waypoints are executed via a low-level PID controller, with replanning occurring every 20 simulation steps ($\sim$ 1 second).

Evaluation Metric. To measure performance, we discretize the top-down maps of each town into a grid of $16m^2$ cells. Total coverage is calculated by summing the area of all unique cells visited over a fixed number of simulation steps. Since slight variations in total path length can occur despite consistent waypoint spacing, we introduce the Area Per Length (APL) metric, defined as the total area covered divided by the total path length. Additionally, we record the percentage of policy failure (PF) as the ratio of the number of agent collisions and the total number of policy calls.

Baselines. We evaluate the exploration performance of VANDERER against the following state-of-the-art methods:

- **NoMaD** [1]: An RGB-only exploration method using a diffusion policy trained on a large-scale navigation dataset (including fine tuning to the place/town data to be deployed in). Hence, we fine-tuned the NoMaD model on our dataset for the same number of epochs.
- **NoMaD***: In our experiments, NoMaD's policy resulted in several collisions. Therefore, we modify it's diffusion architecture to be similar to ours and call this method NoMaD*. Specifically, NoMaD* derives its 1D conditioning vector by pooling encoded patches after the context transformer and replaces the EfficientNet backbone with a pre-trained VAE encoder.
- **DP-RND** [16]: Random Network Distillation (RND) is an intrinsic reward used in RL training to explore unseen states. We define a baseline using RND as a selection metric for our diffusion policy to compare against our VCM.

B. Results

Exploration Performance. A quantitative comparison of VANDERER against the baselines is presented in Table I. Leveraging curiosity-guidance, VANDERER consistently achieves superior exploration performance. While NoMaD demonstrates slightly higher exploration coverage than NoMaD*, it is approximately tenfold more susceptible to policy failures. NoMaD's high policy failure likely arises

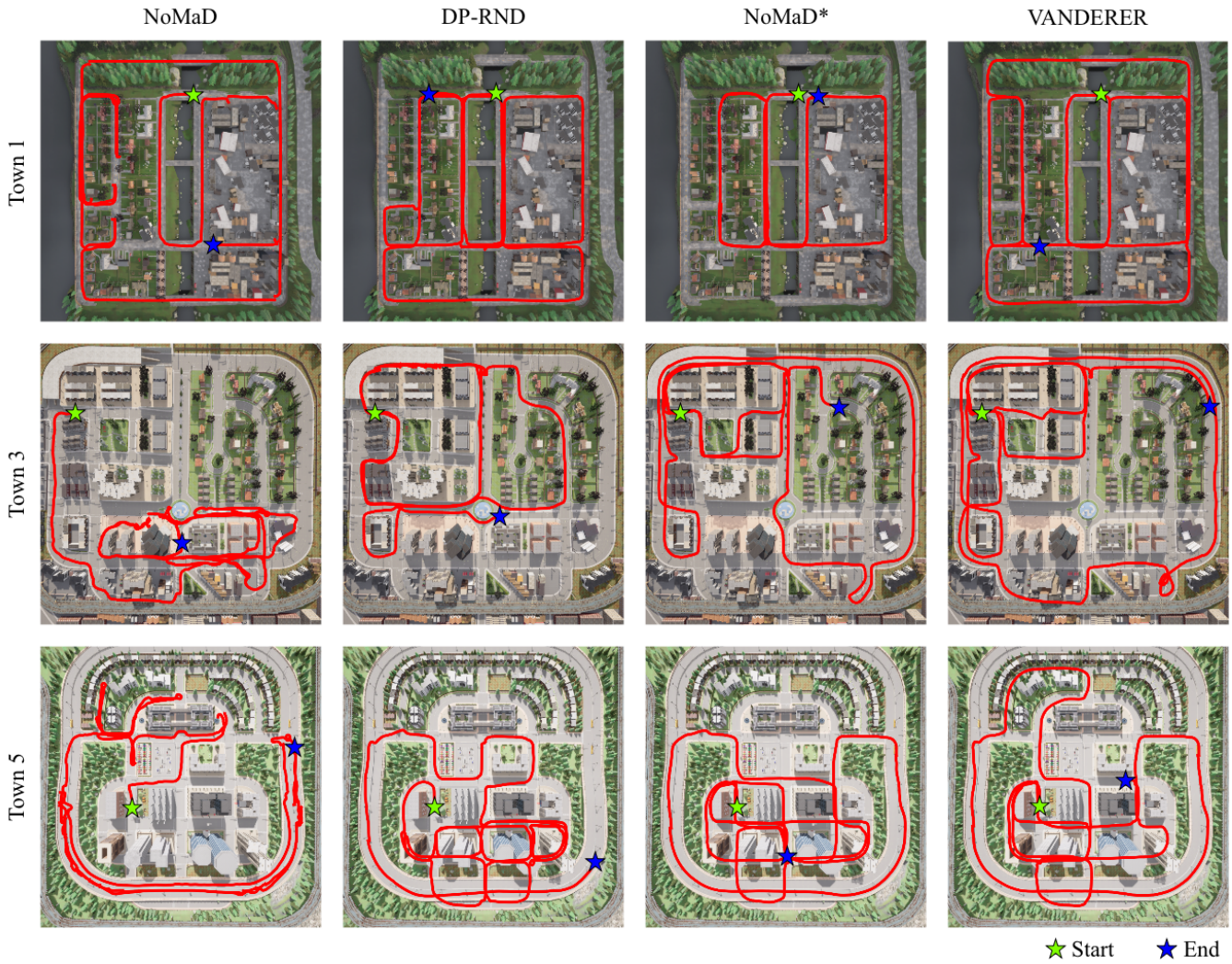


Fig. 4: Qualitative comparison of resultant trajectories across three environments (Town 1, 3, and 5). The red lines illustrate the paths taken by each agent from the **start (green star)** to the **end (blue star)**. While baselines like NoMaD, DP-RND, and NoMaD* often exhibit path repetitions, VANDERER demonstrates more efficient exploration. Notably, NoMaD also shows frequent collisions.

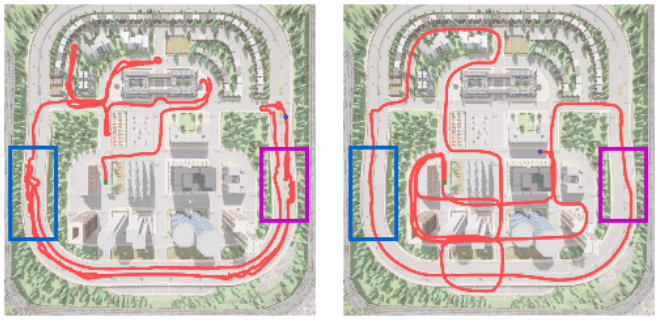
from two critical architectural differences: (1) pooling encoded features before the context transformer, which dilutes fine visual details such as thin poles or raised sidewalks; and (2) the use of the EfficientNet instead of the VAE encoder.

We provide a qualitative comparison of exploration coverage across three environments in Fig. 4. VANDERER consistently achieves superior area coverage across these towns. Furthermore, local trajectory visualizations in Fig. 5 demonstrate that NoMaD's policy leads to frequent collisions, as also indicated by its high PF.

Impact of curiosity-based diffusion guidance. A core contribution of VANDERER is the VCM (curiosity-based diffusion guidance). To evaluate its impact on overall performance, we compare our method against a greedy selection baseline in Fig. 6, where the best-performing action sequence sampled from the diffusion policy is executed without any guidance. As shown in Fig. 6, VANDERER consistently outperforms this baseline across all simulation towns, underscoring the importance of our guidance mechanism.

The greedy baseline faces limitations in scenarios where multiple viable exploratory paths exist, e.g., at an intersection where two directions remain unvisited. Because this approach performs independent optimization at each step τ, it often oscillates between conflicting trajectories at τ and $\tau+1$ rather than committing to a consistent path. Furthermore, the greedy nature of this selection limits performance when immediate candidate actions lead only to previously seen states. In these instances, the lack of stochasticity prevents the agent from discovering unseen locations. In contrast, VANDERER's guidance mechanism avoids such issues by updating the noise distribution rather than greedily selecting the best candidate. This helps in retaining a necessary level of randomness when resampling from the updated noise distribution, allowing the more robust exploration.

Performance with Time. The exploration area is inherently dependent on the number of steps taken by the agent. While all baselines in our experiments are run for a similar number of steps, we also show plots of APL changes as step count



(a) NoMaD [1] **(b)** VANDERER

Fig. 5: The blue and purple boxes highlight two instances of the local paths proposed by policies of NoMaD and VANDERER. NoMaD exhibits frequent collisions, leading to inferior path quality.

Fig. 6: Ablative comparisons of our method with and without diffusion guidance. To show the impact of guidance, we replace it by greedy selection of the candidate with the highest VCM score.

increases (Fig. 7). We compare VANDERER against the previously described greedy selection baseline and a random strategy that selects options randomly at intersections. Initially, when most of the environment is unexplored, all the methods achieve similar APL. However, as the simulation progresses, the performance gap widens, and VANDERER consistently outperforms both baselines.

The comparison between the random and greedy strategies reveals a notable trend: the greedy approach is prone to repetitive looping. When all paths at a junction have been visited, the greedy agent may get trapped in cyclical trajectories. This occasionally results in a drop in APL, causing the greedy baseline to underperform even the random strategy, thus, underscoring the necessity of a guidance-based strategy, such as VANDERER, to maximize exploration efficiency.

Ablation Studies. To evaluate the contribution of the individual components of VANDERER, we conduct ablation studies by systematically removing key architectural elements. Having previously evaluated the impact of the VCM, we now assess the importance of fast reciprocal matching and the diffusion policy. These experiments are conducted in Town 1 of the CARLA dataset, with the metrics in Table II averaged over three runs with distinct starting poses. The results are discussed below:

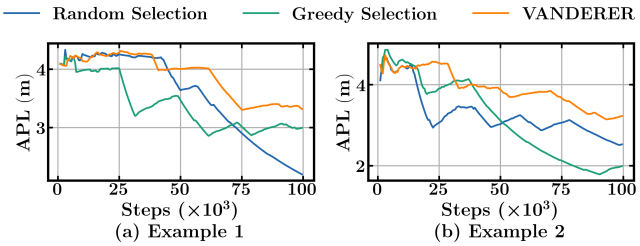


Fig. 7: Performance of Random Selection (NoMaD*), Greedy selection (w/o Guidance), and VANDERER (w Guidance) over simulation steps. (a) Initially, most of the environment is unseen, leading to similar performances from all approaches. However, as steps pass, the impact of VCM and guidance starts becoming evident. (b) Greedy selection can sometimes do repeated exploration in loops, resulting in worse performance than random selection.

- *Without Fast Reciprocal Matching*: We replace the fast reciprocal matching module in the VCM with a direct L_2-norm comparison between corresponding feature patches of the encoded images. Direct pixel-wise comparison underestimates the similarity between two frames showing similar regions but from different viewpoints because identical features appear at different spatial coordinates. On the contrary our reciprocal matching approach effectively accounts for these geometric variances. This is validated by the performance drop observed in Table II.
- *Without Diffusion Policy*: We replace the diffusion policy with a standard Gaussian distribution for sampling action candidates, utilizing CEM for optimization. We maintain identical CEM settings to ensure a fair comparison. The diffusion policy serves as a powerful action prior that significantly accelerates optimization. Ablating this component introduces a fundamental trade-off: a single optimization step (faster computation) yields suboptimal actions and frequent collisions, while increasing optimization iterations renders the method computationally impractical. Consequently, the lack of a strong prior necessitates more intensive optimization, resulting in approximately $4\times$ longer execution time.

TABLE II: Mean performance metrics across three seeds on Town 1. Area: Total Area (m^2), APL: Area per Path Length (m).

	Area	**APL**	**PF**
w/o diffusion policy	7072	1.845	4.35%
w/o fast reciprocal matching	11248	2.856	**0.02%**
Ours	**12299**	**3.120**	0.03%

V. CONCLUSIONS

In settings with limited sensing and computational resources, efficient autonomous exploration remains a significant challenge. To address this, we presented VANDERER, a framework that couples a diffusion policy with a visual-curiosity-based guidance mechanism to enable efficient exploration using only camera data. Our guidance mechanism samples multiple candidate actions and evaluates the novelty

of their predicted outcomes, as estimated by a world model. Rather than employing hard selection, we propose a guidance strategy to better balance exploration and exploitation, resulting in superior APL. Through extensive comparisons, we substantiate that visual curiosity is a powerful metric to drive exploration in sensor-constrained environments. Furthermore, our work highlights the potential for pre-trained diffusion models to be guided toward complex exploratory objectives without the need for specialized expert data.

REFERENCES

[1] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "Nomad: Goal masked diffusion policies for navigation and exploration," in *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2024, pp. 63–70.

[2] B. Yamauchi, "A frontier-based approach for autonomous exploration," *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*, pp. 146–151, 1997.

[3] L. C. Garaffa, M. Basso, A. A. Konzen, and E. P. de Freitas, "Reinforcement learning for mobile robotics exploration: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 3796–3810, 2023.

[4] S. Belkhale, Y. Cui, and D. Sadigh, "Data quality in imitation learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 80 375–80 395, 2023.

[5] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of the Robotics: Science and Systems (RSS)*, 2023.

[6] Y. Cao, J. Lew, J. Liang, J. Cheng, and G. Sartoretti, "DARE: Diffusion Policy for Autonomous Robot Exploration," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2025, pp. 11 987–11 993.

[7] Y. Zeng, H. Ren, S. Wang, J. Huang, and H. Cheng, "Navidiffusor: Cost-guided diffusion model for visual navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2025, pp. 11 994–12 001.

[8] A. Bar, G. Zhou, D. Tran, T. Darrell, and Y. LeCun, "Navigation world models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 15 791–15 801.

[9] L. Heng, A. Gotovos, A. Krause, and M. Pollefeys, "Efficient visual exploration and coverage with a micro aerial vehicle in unknown environments," in *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2015, pp. 1071–1078.

[10] H. Umari and S. Mukhopadhyay, "Autonomous robotic exploration based on multiple rapidly-exploring randomized trees," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 1396–1402.

[11] B. Lindqvist, A.-A. Agha-Mohammadi, and G. Nikolakopoulos, "Exploration-rrt: A multi-objective path planning and exploration framework for unknown and unstructured environments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2021, pp. 3429–3435.

[12] S. M. LaValle, "Rapidly-exploring random trees : a new tool for path planning," *The annual research report*, 1998. [Online]. Available: https://api.semanticscholar.org/CorpusID:14744621

[13] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[14] M. Selin, M. Tiger, D. Duberg, F. Heintz, and P. Jensfelt, "Efficient autonomous exploration planning of large-scale 3-d environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1699–1706, 2019.

[15] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 06–11 Aug 2017, pp. 2778–2787.

[16] Y. Burda, H. Edwards, A. J. Storkey, and O. Klimov, "Exploration by random network distillation," in *Proceedings of the 7th International Conference on Learning Representations*, 6-9 May 2019.

[17] D. Pathak, D. Gandhi, and A. Gupta, "Self-supervised exploration via disagreement," in *Proceedings of the International Conference on Machine Learning*. PMLR, 2019, pp. 5062–5071.

[18] Y. Cao, T. Hou, Y. Wang, X. Yi, and G. Sartoretti, "Ariadne: A reinforcement learning approach using attention-based deep networks for exploration," in *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2023, pp. 10 219–10 225.

[19] H. Li, Q. Zhang, and D. Zhao, "Deep reinforcement learning-based automatic exploration for navigation in unknown environment," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 6, pp. 2064–2076, 2019.

[20] X. Chen, T. Wang, Q. Li, T. Huang, J. Pang, and T. Xue, "Gleam: Learning generalizable exploration policy for active mapping in complex 3d indoor scene," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 5558–5568.

[21] Y. Cao, R. Zhao, Y. Wang, B. Xiang, and G. Sartoretti, "Deep reinforcement learning-based large-scale robot exploration," *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4631–4638, 2024.

[22] A. H. Tan, S. Narasimhan, and G. Nejat, "4cnet: A diffusion approach to map prediction for decentralized multi-robot exploration," *IEEE Transactions on Robotics*, 2026.

[23] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "Vint: A foundation model for visual navigation," in *Proceedings of The 7th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Tan, M. Toussaint, and K. Darvish, Eds., vol. 229. PMLR, 06–09 Nov 2023, pp. 711–733.

[24] D. Shah, B. Eysenbach, N. Rhinehart, and S. Levine, "Rapid exploration for open-world navigation with latent goal models," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 164. PMLR, 08–11 Nov 2022, pp. 674–684.

[25] A. Nayak, D. O. Makowski, S. Gode, C. Schmid, and W. Burgard, "Metricnet: Recovering metric scale in generative navigation policies," *arXiv preprint arXiv:2509.13965*, 2025.

[26] S. Gode, A. Nayak, D. N. Oliveira, M. Krawez, C. Schmid, and W. Burgard, "Flownav: Combining flow matching and depth priors for efficient navigation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2025, pp. 17 762–17 768.

[27] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.

[28] J. Ho and T. Salimans, "Classifier-free diffusion guidance," in *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[29] X. Guo, J. Liu, M. Cui, J. Li, H. Yang, and D. Huang, "Initno: Boosting text-to-image diffusion models via initial noise optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 9380–9389.

[30] C. Chen, L. Yang, X. Yang, L. Chen, G. He, C. Wang, and Y. Li, "Find: Fine-tuning initial noise distribution with policy optimization for diffusion models," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 6735–6744.

[31] K. Karunratanakul, K. Preechakul, E. Aksan, T. Beeler, S. Suwajanakorn, and S. Tang, "Optimizing diffusion noise can serve as universal motion priors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 1334–1345.

[32] H. et.al, "Constraint-aware diffusion guidance for robotics: Real-time obstacle avoidance for autonomous racing," in *Proceedings of The 9th Conference on Robot Learning*, J. Lim, S. Song, and H.-W. Park, Eds., vol. 305. PMLR, 27–30 Sep 2025, pp. 1756–1776.

[33] B. et. al, "Stable video diffusion: Scaling latent video diffusion models to large datasets," *arXiv preprint arXiv:2311.15127*, 2023.

[34] V. Leroy, Y. Cabon, and J. Revaud, "Grounding image matching in 3d with mast3r," in *Proceedings of the European Conference on Computer Vision*. Springer, 2024, pp. 71–91.

[35] R. G. Goswami, A. Bar, D. Fan, T.-Y. Yang, G. Zhou, P. Krishnamurthy, M. Rabbat, F. Khorrami, and Y. LeCun, "World models can leverage human videos for dexterous manipulation," *arXiv preprint arXiv:2512.13644*, 2025.

[36] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., vol. 78. PMLR, 13–15 Nov 2017, pp. 1–16.