SIGN: Safety- ware Image-Goal Navigation for utonomous Drones via Reinforcement Learning

Zichen Yan, Rui Huang, Lei He, Shao Guo, Lin Zhao

***bstract*—Image-goal navigation (ImageNav) tasks a robot with autonomously exploring an unknown environment and reaching a location that visually matches a given target image. While prior works primarily study ImageNav for ground robots, enabling this capability for autonomous drones is substantially more challenging due to their need for high-frequency feedback control and global localization for stable flight. In this paper, we propose a novel sim-to-real framework that leverages reinforcement learning (RL) to achieve ImageNav for drones. To enhance visual representation ability, our approach trains the vision backbone with auxiliary tasks, including image perturbations and future transition prediction, which results in more effective policy training. The proposed algorithm enables end-to-end ImageNav with direct velocity control, eliminating the need for external localization. Furthermore, we integrate a depth-based safety module for real-time obstacle avoidance, allowing the drone to safely navigate in cluttered environments. Unlike most existing drone navigation methods that focus solely on reference tracking or obstacle avoidance, our framework supports comprehensive navigation behaviors, including autonomous exploration, obstacle avoidance, and image-goal seeking, without requiring explicit global mapping. Code and model checkpoints are available at https://github.com/Zichen-Yan/SIGN.**

***Index Terms*—Image-goal navigation, reinforcement learning, obstacle avoidance, autonomous drones.**

I. Introduction

L E RNING vision-based navigation has attracted increasing attention in autonomous drone research. Existing works mostly focus on achieving autonomous obstacle avoidance and reference tracking, where the reference signal can be a target position [1], velocity [2]–[4] or direction [5], [6]. ImageNav poses a further challenge: enabling autonomous drones to search an unknown environment for a given goal image while avoiding collisions. This problem remains largely unexplored but has significant value in practical applications such as disaster rescue and industrial inspection, especially when pre-built maps and external localization are unavailable. Purely vision-based navigation reduces the dependence on additional sensors to enable a low-cost and lightweight drone design. In this paper, we investigate ImageNav for autonomous drones using only visual and IMU measurements.

The body of ImageNav research has primarily focused on ground robots [7]–[10]. Most existing studies are conducted exclusively in simulators such as Habitat [11] and neglect to penalize collisions with obstacles, which can result in unsafe behaviors when deployed in the real world. Meanwhile, discrete action spaces such as $\{$*move_forward*, *turn_left*, *turn_right*, *stop*$\}$ are commonly adopted to simplify learning by reducing the search space. This is generally only acceptable for ground robots, which are inherently stable and can correct their position through vision by the discrete low-frequency actions.

In contrast, achieving safe and efficient ImageNav for autonomous drones is considerably more challenging. Drones are inherently unstable platforms with fast dynamics, necessitating high-frequency feedback control and accurate global localization to maintain stable flight. When relying on visual-inertial localization, they are particularly susceptible to pose-estimation drift, especially in indoor environments. Furthermore, discrete-action policies often lack the precision and responsiveness needed for rapid pose correction. For flight control systems, frequent switching between discrete actions can severely degrade real-time tracking performance due to the abrupt change of references. In this case, a continuous policy that directly outputs velocity commands can address this issue by enabling smoother and more timely control. Continuous actions also support seamless integration with flight controllers without additional planning modules. However, it also expands the exploration space, substantially increasing training complexity. In addition, naively embedding safety constraints into policy learning can hinder effective exploration and degrade navigation performance, particularly in cluttered or narrow environments such as Gibson [12], MP3D [13], and HM3D [14]. Therefore, balancing efficient exploration with robust real-world safety remains a central challenge in autonomous drone navigation.

In this paper, we introduce **SIGN** (**S**afety-aware **I**mage-**G**oal **N**avigation), an end-to-end RL-based ImageNav framework for autonomous drones that employs continuous velocity control and incorporates a safety module for collision avoidance. Our main contributions are as follows:

- SIGN performs end-to-end ImageNav on real drones using continuous velocity control, with only visual and IMU inputs and does not rely on external localization. It supports seamless sim-to-real transfer without any fine-tuning and is validated on both simulation benchmarks and real-world experiments.
- We propose self-supervised auxiliary tasks that significantly improve training efficiency and achieve state-of-the-art performance on the Gibson benchmark under continuous control. These tasks can be seamlessly integrated into on-policy RL training. More broadly, SIGN offers a general framework for extending existing discrete-action

This paper is supported by the Singapore Ministry of Education Tier 2 cademic Research Funds (T2EP20123-0037, T2EP20224-0035). (Corresponding author: Lin Zhao) The authors are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. Email: zicheny n@u.nus.edu, ruihu ng@u.nus.edu, lei.he@nus.edu.sg, e1374427@u.nus.edu, zh olin@nus.edu.sg

Manuscript received ugust 18, 2025; accepted December 1, 2025.

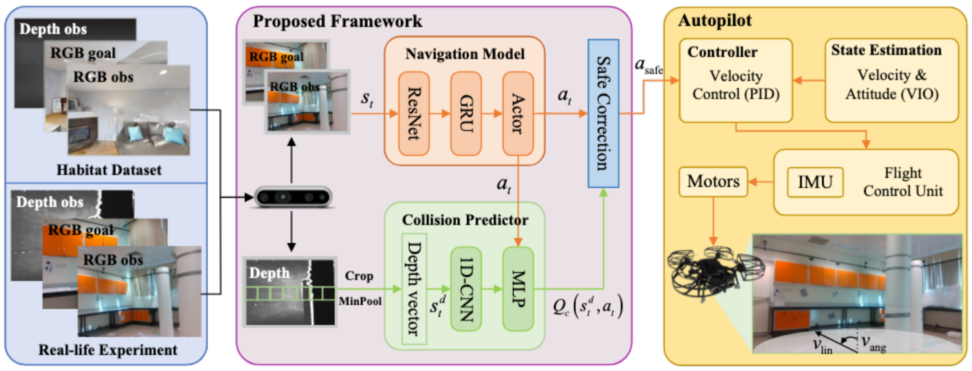

Fig. 1. SIGN enables end-to-end ImageNav for autonomous drones via visual reinforcement learning with effective sim-to-real transfer. Guided by the collision predictor, an action correction mechanism is incorporated to ensure obstacle-avoiding navigation in indoor environments.

ImageNav algorithms to continuous control settings.

- Moreover, while existing end-to-end ImageNav algorithms often fail to account for safety, we introduce a safety module that predicts collision probability from depth inputs and corrects potentially unsafe actions, thereby ensuring reliable obstacle avoidance.

II. REL TED WORKS

. *Visual Navigation on Drones*

We focus on end-to-end, map-less vision-based navigation without relying on global localization. Prior research in this direction has mainly addressed reference-tracking control with obstacle avoidance [1]–[5] or drone racing on a fixed course [15], where an RL policy maps sensor inputs directly to control commands such as velocity or acceleration. The key distinction from these tasks is that ImageNav specifies the reference signal through an image, which serves as a visual goal for navigation. Similarly, Tao [16] explored RL-based indoor search using drones, but their approach depends on global mapping. In contrast, our task is more challenging as it requires not only map-less exploration and collision avoidance, but also navigation toward a specified goal image.

B. *End-to-end RL-based Visual Navigation*

For visual navigation tasks, end-to-end RL has demonstrated higher efficiency in real-time inference and module tuning by avoiding intermediate stages such as mapping and subgoal selection [17]. Early work by Wijmans [18] addressed point-goal visual navigation for ground robots. However, it relied on GPS and compass data, which is impractical for indoor settings. l-Halah [7] removed the dependency on localization and proposed a map-less visual navigation framework, enabling zero-shot generalization to new tasks and multi-modal goals. dditionally, previous works have employed memory mechanisms [19] and topological maps [20]–[22] to enhance exploration efficiency by storing keyframes and leveraging graph neural networks to extract visual correlations for guidance. For now, most existing end-to-end RL methods are constrained to discrete actions and suffer from training inefficiency due to environmental diversity. To better facilitate drone deployment and simplify the sim-to-real transfer, we investigate the extension from discrete to continuous settings.

C. *Image-conditioned Waypoint Planner*

Many works predict waypoints to reach a target image and use low-level position controllers for trajectory tracking [23]–[25], even assisted by a world model [26]. common weakness of these methods lies in the out-of-distribution (OOD) issues inherent to the supervised learning paradigm, especially when the current and target images have no overlap. In many image-conditioned navigation methods, continuous policies have already been explored through imitation learning. However, ImageNav involves numerous OOD scenarios that make generalization challenging without global localization to guide exploration. To address this issue, an RL-based continuous policy presents a promising solution. dditionally, an emerging trend is to combine 3D reconstruction techniques with ImageNav for drones, such as NeRF [27] and Gaussian Splatting [28]. They typically require offline reconstruction and substantial computational resources. Due to the limited onboard capacity of drones, real-time processing of such methods remains impractical.

III. VISU L REINFORCEMENT LE RNING FOR IM GEN V

s depicted in Fig. 1, SIGN is an end-to-end framework for drone ImageNav that leverages a continuous policy

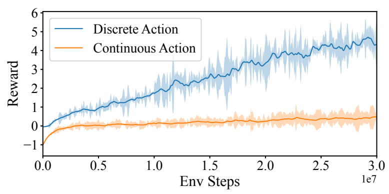

Fig. 2. Performance comparison between discrete and continuous action using the SoT baseline FGPrompt [8].

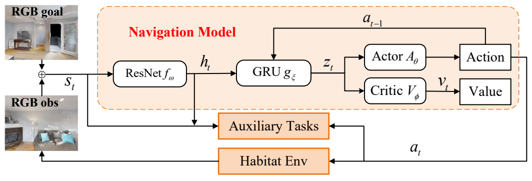

Fig. 3. Overview of the navigation model.

to enable map-less navigation with enhanced safety. But training continuous policies for ImageNav presents a significant challenge due to the expanded action space and increased training complexity. In Fig. 2, we compare a discrete State-of-The- rt (SoT) baseline with its continuous counterpart, where the discrete action space $\mathcal{A}_d = \{move\_forward, turn\_left, turn\_right, stop\}$ and the continuous one consists of forward linear and angular velocities, $\mathcal{A}_c = \{v_{lin}, v_{ang}\}$. It can be observed that the discrete-action policy achieves steadily increasing rewards and much higher training efficiency, while the continuous-action policy suffers from very slow reward growth. This highlights the difficulty of training continuous policies in ImageNav tasks and motivates the design of our SIGN framework to address these challenges.

. Task Definition

Given a goal RGB image in an unknown environment, the agent is required to search the area for the target based on first-person observations from a single camera. t each step t, the agent samples an action a_t conditioned on visual observations s_t, and gets a reward r_t. The task is considered successful if the distance to the goal is less than 1 m and the angle between the target image and current image is less than $25°$.

B. Navigation Model

The navigation model consists of a ResNet encoder f_ω, a GRU model g_ξ, and an actor-critic policy trained by Proximal Policy Optimization (PPO) [29]. See Fig. 3 for the diagram. The goal image O_g and current image O_c are stacked as the input s_t. The inference procedure can be summarized as

$$\begin{aligned} s_t &= O_c \oplus O_g, \\ h_t &= f_\omega(s_t), z_t = g_\xi(h_t, a_t \;\; 1), \\ a_t &\sim A \;\; (\cdot|z_t), v_t = V_\phi(z_t). \end{aligned} \tag{1}$$

C. ugmented Continuous Policy Training

possible reason of the inefficient continuous policy training in Fig. 2 is that, in the early stage, training an effective vision encoder from scratch is time-consuming. RL policy struggles to improve until the vision encoder learns to capture key information from pixels. To address this problem, we propose two useful Self-Supervised Learning (SSL) techniques as auxiliary tasks to enhance the policy learning.

Recall the standard optimization form in vanilla PPO [29]

$$\begin{aligned} &\max L_a(\theta) = \;_t\left[\min(r_t(\theta)\hat{A}_t, \mathrm{clip}(r_t(\theta), 1 \;\; \epsilon, 1+\epsilon)\hat{A}_t)\right] \\ &\min_\phi L_v(\phi) = \;_t\left[\frac{1}{2}(V_{gt} \;\; V_\phi(s_t))^2\right] \end{aligned} \tag{2}$$

where $\hat{A}_t$ adopts generalized advantage estimation and $r_t(\theta) = \pi \;\; (a_t|s_t)/\pi \;\; _{ol} \;\; (a_t|s_t)$. uxiliary tasks are built on the PPO structure and aim to robustify policy and enhance representation ability by learning to predict future states.

1) Future Prediction: The prediction ability receives widespread attention due to its success in the model-based visual RL research [26], [30], where a foundation world model acts as an internal simulator to imagine future observations and guide policy learning. Focusing on a portable model for drone applications, here we train a single-step prediction to enhance the visual backbone training. To this end, future prediction task is introduced, as illustrated in Fig. 4. We deploy an MLP head P_ψ after the visual encoder f_ω to predict the reward r_t, the next-step latent feature $h_{t+1} = f_\omega(s_{t+1})$, and the done flag d_t. The auxiliary training loss is defined as

$$\begin{aligned} L_{fp}(\omega, \psi) = \lambda_r &\underbrace{\mathrm{CrossEntropy}(\hat{r}_t, \text{Two-hot}(r_t))}_{\text{reward loss}} \\ &+ \lambda_d \underbrace{\mathrm{MSE}(\hat{h}_{t+1}, h_{t+1})}_{\text{dynamic loss}} + \lambda_T \underbrace{\mathrm{MSE}(\hat{d}_t, d_t)}_{\text{termination loss}}. \end{aligned} \tag{3}$$

The forward prediction of dynamics $\hat{h}_{t+1} = P_\psi(a_t, h_t)$ is implemented in the latent space to avoid deconvolution operations and save computation resources. The reward loss function uses cross-entropy between the predicted reward $\hat{r}_t$ and a two-hot encoding of reward r_t, which proves more effective in predicting sparse rewards and robust to reward magnitude [31].

2) RandomShift: The RandomShift [32] technique performs data augmentation on the current image O_c in each sample batch. Denote the perturbated data as $aug(s_t) = randomshift(O_c) \oplus O_g$, then auxiliary loss is defined as

$$\begin{aligned} L_{rs}(\theta, \phi) = \mathrm{KL}(&\underbrace{\pi \;\; (\cdot|s_t)}_{\text{detach}}, \pi \;\; (\cdot|aug(s_t))) \\ &+ \mathrm{MSE}(\underbrace{V_\phi(s_t)}_{\text{detach}}, V_\phi(aug(s_t))), \end{aligned} \tag{4}$$

where 'detach' operation is applied to block gradients. The purpose of RandomShift is to maintain the action distribution and value prediction when dealing with slightly shifted images. It helps improve the model's robustness in the presence of jitter in real-life images. The goal image O_g remains unchanged in

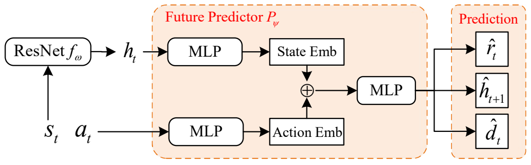

Fig. 4. uxiliary task to enhance the vision encoder f by predicting the reward, future latent state, and done flag.

lgorithm 1 ugmented Continuous Policy Training

```
1:  Input: number of env $B$, trajectory length $L$, fixed weights $\lambda_r, \lambda_d, \lambda_T, \lambda_{rs}$
2:  Initialize actor-critic models $A$ and $V_\phi$ including the backbone $f_\omega$ and GRU $g_\xi$
3:  Initialize predictor model $P_\psi$ and collision predictor $Q_c$
4:  for each epoch do
5:      Collect one trajectory $\{s_t, a_t, s_{t+1}, r_t, c_t, d_t\}_{1:L}$ from each env, a total of $B \times L$ samples
          # Future Prediction
6:      for each mini-batch do
7:          Update $P_\psi$ and $f_\omega$ with data $\{s_t, a_t, s_{t+1}, r_t, d_t\}$ via minimizing Eq. (3)
8:      end for
          # PPO training with RandomShift
9:      for each mini-batch do
10:         Update policy parameters $\omega, \xi, \theta, \phi$ via Eq. (5)
11:         Train collision predictor $Q_c$ via minimizing Eq. (9)
12:     end for
13: end for
```

auxiliary training to align with the reward distribution. Finally, the actor-critic network can be updated with RandomShift. By adding Eq. (4) into the PPO update rule, the augmented optimization becomes

$$\min_{,\phi} \quad L_a + L_v + \lambda_{rs} L_{rs} \tag{5}$$

The whole training process is shown in lgorithm 1.

IV. S FETY MODULE FOR RE L-WORLD EXPERIMENTS

Existing ImageNav algorithms primarily focus on navigation performance while neglecting safety. Navigation policies trained in Habitat in those works are allowed to collide with obstacles. In practice, enforcing safety during policy training can lead to conservative behavior, which significantly hinders exploration in narrow indoor environments. Inspired by the safety filter designs in the fields of safe RL and control [33]–[35], we propose a practical action correction mechanism that is activated only when the agent is in danger. This decoupled design is shown to be effective in balancing exploration and safety. Meanwhile, it also simplifies training.

. *Model Training for Collision Probability Prediction*

The depth image is first preprocessed by cropping and min-pooling operations to obtain a depth vector $s_t^d \in \mathbb{R}^n$, indicating the closest distance in different directions. Q_c is used to predict the collision probability based on the current depth image and navigation action a_t, defined as

$$Q_c(s_t^d, a_t) = \quad _t[c_t | s_t^d, a_t], \quad c_t = \begin{cases} 1, & \text{if collision,} \\ 0, & \text{else.} \end{cases} \tag{6}$$

where c_t is a collision feedback from the Habitat simulator.

However, our experimental results show that using only hard labels c_t leads to poor prediction, as the predicted probability does not vary smoothly with distance and lacks a gradual transition from safe to dangerous states. To obtain more reliable probability estimates, we introduce soft labels c_t^{soft} that provide a smoother and more accurate indication of collision likelihood:

$$c_t^{\text{soft}} = \text{Sigmoid} \quad 10 \cdot (\beta \quad \min_i s_t^d(i))\Big), \tag{7}$$

where $\min(\cdot)$ denotes the minimum element of vector s_t^d and β is the threshold for a safe distance. When the minimum perceived distance falls below β, the soft label value increases rapidly toward 1, and it decreases to 0 when the distance exceeds β. Thus, collision detection is defined with respect to a safety margin distance of β.

The collision predictor Q_c is trained using a 1:1 mixture of hard and soft labels. The final loss function L_c becomes

$$\text{Label}_{\text{mix}} = \text{Mix}(c_t, c_t^{soft}), \tag{8}$$

$$L_c = \text{BCELoss}(\hat{Q}_c(s_t^d, a_t), \text{Label}_{\text{mix}}). \tag{9}$$

The benefits of mixed labels are two-fold: 1) Soft labels c_t^{soft} enable smooth predictions since they contain continuous depth information; 2) Hard labels help to learn the correlation between action a_t and the collision probability $\hat{Q}_c(s_t^d, a_t)$, facilitating the following action correction mechanism. s is shown in lgorithm 1, Q_c can be trained jointly with the RL policy in the Habitat simulator.

B. *ction Correction for Obstacle voidance*

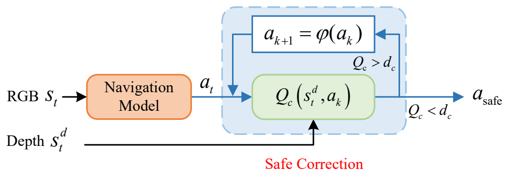

Fig. 5. correction mechanism for navigation actions to enhance safety.

The predicted collision probability is conditioned on s_t^d and a_t, which means we can modify the action to reduce the collision probability. The correction mechanism is illustrated in Fig. 5. When the value of Q_c exceeds a pre-defined threshold d_c, it indicates a dangerous situation, then actions from the navigation policy will be corrected iteratively until $Q_c < d_c$. The correction function $\varphi(\cdot)$ will project the raw action $a_0 = a_t$ to a safer one based on gradients [33]

$$a_{k+1} = \varphi(a_k) = a_k \quad \frac{\partial}{\partial a_k}[Q_c(s_t^d, a_k) \quad d_c]^+, \tag{10}$$

where $k = 0, 1, \ldots, n$ is the iteration number. The adjustment of the probability threshold d_c can be viewed as a trade-off between safety and navigation performance. In general cases, d_c is set to 0.5 because when $Q_c > 0.5$, the minimum distance is less than the safety distance β according to Eq. (7).

V. EXPERIMENTS

This section aims to answer the following key questions:

- Can SIGN enable efficient training of continuous ImageNav policies? Do the proposed auxiliary tasks contribute to increased sample efficiency?

- How to achieve sim-to-real transfer from the Habitat simulator to real drones?
- Is the safety module able to ensure obstacle avoidance during visual navigation?

. Simulation Settings

1) Datasets: Our navigation model is trained on Gibson ImageNav dataset [12], including 72 training scenes and 14 testing scenes. To further validate the robustness of our algorithm, the trained model is also evaluated on cross-domain datasets MP3D [13] and HM3D [14]. Scenes from different datasets have varying backgrounds and layouts but are all simulated in the Habitat.

2) gent Configurations: To be consistent with SoT works, we employ the Habitat camera that has a $90°$ Field of View (FoV) and 128×128 resolution. The continuous action space $\mathcal{A}_c$ consists of forward linear velocity $v_{lin} \in [0, 0.25]$ m/s and angular velocity $v_{ang} \in$ $[\ \ 15, 15]$ $°/s$. To implement the *'Stop'* action in a continuous setting, the episode will be terminated early when $v_{lin} <$ 0.025 m/s & $|v_{ang}| < 1.5$ $°/s$.

3) Reward and Performance Metrics: The design of the reward function r_t follows ZER [7], which encourages the policy to approach the target and reduce the angle between the current view and the view of the goal.

$$r_t = r_{dis}(d_t, d_{t\ \ 1}) + [d_t \leq d_s][\ \ _t \leq \ \ _s]r_{suc} + \gamma, \tag{11}$$

where d_t denotes the distance to the target position at time t and $r_{dis}(d_t, d_{t\ \ 1}) = d_{t\ \ 1} \quad d_t$ is the approaching reward. $_t$ indicates the angle between the current view and the target view. $[\cdot]$ is defined as an indicator function

$$[x] = \begin{cases} 1, & \text{if } x \text{ is True,} \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

In our experiments, $d_s = 1$ m and $\ \ _s = 25°$ denote the success thresholds for distance and angular deviation, respectively. $r_{suc} = 2.5$ is the success reward and $\gamma = \ \ 0.01$

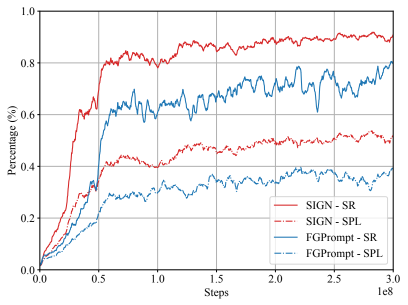

Fig. 6. Comparison of training curves between SIGN and the continuous baseline.

is the slack penalty. On benchmarks, Success Rate (SR) and Success Weighted by Path Length (SPL) are two main metrics to quantify the navigation performance. SPL is calculated as $\frac{1}{N}\Sigma_{i=1}^{N} S_i \frac{L_i}{max\ \ L_i, p_i)}$, where N is the total number of episodes, $S_i \in \{0, 1\}$ is a success indicator, p_i is the actual path length in the i-th episode, and L_i is the optimal path length from the start to the goal point.

4) Training Details: To enable fast inference and hardware-friendly deployment, SIGN uses a lightweight backbone ResNet9. The whole model contains 1.6M parameters and is trained from scratch for 300M steps using an RTX 4090 GPU. PPO adopts $B = 8$ parallel environments for distributed training on the Gibson benchmark. For the rest of the hyperparameters in lgorithm 1, we choose $\lambda_d = 1, \lambda_r = 0.1, \lambda_T = 0.1, \lambda_{rs} = 0.5$ (loss weights, see Eq.(3) and (5)), and $L = 256$.

B. Performance nalysis on Benchmarks

1) Training Results: The comparison between SIGN and other SoT algorithms is presented in Table I. Due to the lack of existing continuous baselines, we extend FGPrompt (a discrete SoT method) to a continuous policy by modifying its action layer. To ensure a fair comparison, both continuous FGPrompt and SIGN are trained for the same number of steps, as illustrated in Fig. 6. On the Gibson benchmark, SIGN outperforms the continuous baseline by **+6.4%** in SR and **+18.3%** in SPL. With the augmentation of the proposed auxiliary tasks, SIGN demonstrates better training efficiency than the baseline.

The performance gap between discrete and continuous policies arises from the increased task complexity. Specifically, discrete policies simplify system dynamics by updating the agent's pose through teleportation. In contrast, continuous velocity control follows first-order integral dynamics ($\dot{p} = v$), requiring the agent to reason over continuous transitions and jointly optimize linear and angular velocities. s a result, training continuous-control policies is more challenging for RL, especially for early exploration and policy convergence. The significant performance degradation of FGPrompt when transferred from discrete to continuous actions further highlights the challenges of continuous control in ImageNav.

To further evaluate the generalization ability, algorithms trained on Gibson are directly transferred to unseen scenarios

T BLE I
COMP RISON WITH SOT METHODS ON GIBSON.

Method	Backbone	Sensor(s)	ction Space	SPL( )	SR( )
Mem- ug [19]	ResNet18	4 RGB	Discrete	56.0%	69.0%
VGM [20]	ResNet18	4 RGB	Discrete	64.0%	76.0%
TSGM [21]	ResNet18	4 RGB	Discrete	67.2%	81.1%
ZER [7]	ResNet9	RGB	Discrete	21.6%	29.2%
ZSON [10]	ResNet50	RGB	Discrete	28.0%	36.9%
OVRL-V2 [9]	ViT-Base	RGB	Discrete	58.7%	82.0%
FGPrompt [8]	ResNet9	RGB	Discrete	66.5%	90.4%
FGPrompt [8]	ResNet9	RGB	Continuous	35.0%	79.9%
SIGN (Ours)	ResNet9	RGB	Continuous	53.3%	86.3%

T BLE II
CROSS-DOM IN EV LU TION ON MP3D ND HM3D.

Methods	ction Space	MP3D		HM3D	
		SPL( )	SR( )	SPL( )	SR( )
Mem- ug [19]	Discrete	3 9	6 9	3 5	1 9
NRNS [22]	Discrete	5 2	9 3	4 3	6 6
ZER [7]	Discrete	10 8	14 6	6 3	9 6
FGPrompt [8]	Discrete	50 4	77 6	49 6	76 1
FGPrompt [8]	Continuous	22 9	59 4	20 1	56 5
SIGN (Ours)	Continuous	35 0	65 4	30 9	64 5

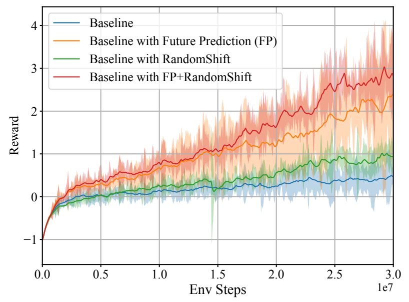

Fig. 7. blation results of training continuous polices with proposed auxiliary tasks.

on MP3D and HM3D datasets. The results are averaged across easy, medium, and hard modes. Each mode contains 1000 episodes. The evaluation results on the cross-domain datasets are shown in Table II. SIGN outperforms continuous FGPrompt by **+6%** SR and **+12.1%** SPL on MP3D, and **+8%** SR and **+10.8%** SPL on HM3D. Overall, SIGN is thoroughly validated across diverse ImageNav benchmarks and achieves SoT in the continuous setting.

2) blation Study: To evaluate the effectiveness of two auxiliary tasks in the policy training, ablation experiments with different random seeds are carried out on Gibson benchmark for 30M steps. The results are shown in Fig. 7. Compared to the baseline, the incorporation of either SSL technique accelerates training and enhances performance during the early stages. Combining both with PPO achieves the best results.

C. Reliability nalysis of Safety Module

Obstacle avoidance is guided by the collision predictor Q_c that is trained together with the navigation policy. To further evaluate its robustness and joint performance with the PX4 flight control system, repeated obstacle avoidance experiments were conducted in Gazebo with pole obstacles. The simulation environment is provided by XTDrone [36], as shown in Fig. 8(a). Gazebo simulator can be integrated with PX4 to closely approximate real-life flight dynamics.

We define a set of start and goal locations, and for each start-goal pair, 10 repeated point-to-point flight trials are conducted. The PX4 controller receives reference commands: angular velocity v_{ang} and forward linear velocity v_{lin}. The reference v_{lin} is set as $0.25\ m/s$, which matches the maximal speed

TABLE III
RESULTS OF OBST CLE VOID NCE EXPERIMENTS.

Start-End Pair	SPL( )	SR( )	Correction Rate
S1-E1	95.8%	100%	8.9%
S1-E2	96.6%	100%	7.0%
S1-E3	88.7%	100%	17.0%
S2-E1	83.2%	90%	14.6%
S2-E2	96.4%	100%	7.2%
S2-E3	76.2%	90%	23.9%
S3-E1	96.9%	100%	6.1%
S3-E2	85.5%	100%	19.6%
S3-E3	85.4%	90%	10.4%
verage	89.4%	96.7%	12.7%

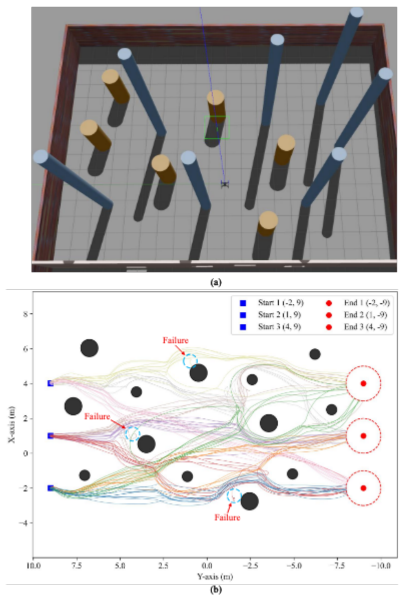

Fig. 8. Results of reliability tests for obstacle avoidance. (a) Evaluation environment in Gazebo. (b) Flying trajectories from each start point to the goal point.

during ImageNav. The reference v_{ang} is computed based on the yaw deviation between the current orientation and the direction toward the goal. In Fig. 8(b), the blue square indicates the start point, the red circle denotes the end point, and black disks are obstacles. trial is considered successful if the drone reaches within $1\ m$ of the goal without any collision. This setup allows us to evaluate the effectiveness of the proposed action correction mechanism.

In our experiments, depth perception is limited to 3 m, and we denote the normalized depth by s_t^d with $0 \leq s_t^d(i) \leq 1$, corresponding to a real-world range of $[0, 3]$ m. ccordingly, the hyperparameter $\beta = 0.3$ in Eq.(7) establishes a 0.9 m safety margin, chosen based on the maximal flight speed. Each depth image is cropped to the central 20% horizontal strip, divided into 16 blocks, and the minimum value from each block forms $s_t^d \in \mathbb{R}^{16}$, which is then fed to Q_c. In Eq. (10), the collision probability threshold is set to $d_c = 0.5$, corresponding to the Sigmoid function's decision boundary, and actions are iteratively corrected to ensure $Q_c < d_c$.

In our tests, the collision predictor Q_c tends to reduce the collision probability by slowing down v_{lin} and increasing turning speed v_{ang}. In some cases, gradient-based correction in Eq. (10) may induce oscillatory motions, as alternating updates to the angular velocity can cause the drone to swing laterally. To mitigate this issue, a more practical way is to apply action correction iteratively at a fixed interval

$$\varphi(a_k) = a_k \qquad a = (v_{lin} \qquad _{lin}, v_{ang} \quad D \cdot \quad _{ang}), \quad (13)$$

where $D \in \{\ \ 1, 1\}$ denotes a precomputed safer direction based on the current depth vector s_t^d. For a normalized

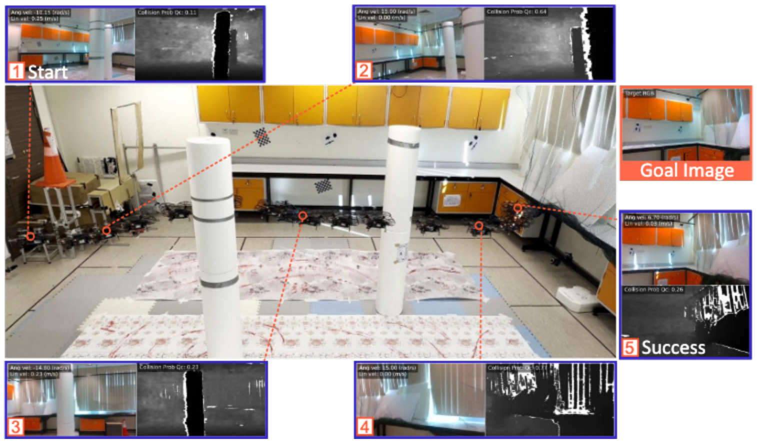

Fig. 9. Flight trajectory during the ImageNav task. The image within the red box represents the goal. The intermediate observations, actions, and predicted collision probabilities can be found on the images within the blue boxes.

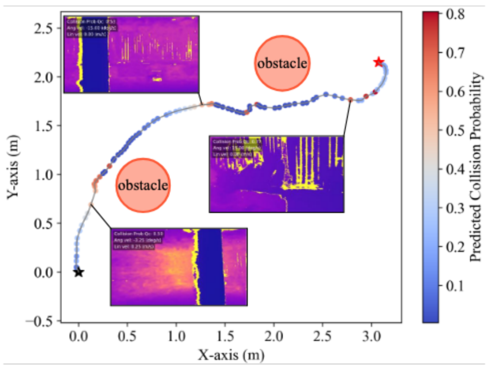

Fig. 10. First-person-view depth images at key waypoints for obstacle avoidance. Waypoints are color-coded by their predicted collision probability Q_c, with warmer colors indicating higher risk. The trajectory starts at the black star and terminates at the red star.

$v_{lin}, v_{ang} \in [\ 1, 1]$, we set $\ _{lin} = 0.3$, $\ _{ang} = 0.3$ for both simulations and real-life experiments. To ensure real-time inference, we limit the number of corrections to a maximum of five. If the predicted collision probability is still larger than d_c after five corrections, the forward speed will be reset to 0. Then the drone will rotate in place at a speed of v_{ang} to escape from local dead ends.

The experimental results are presented in Table III and Fig. 8(b). For each start-end pair, we compute the average metrics across 10 trials, resulting in a total of 90 trajectories. The overall SR is 96.7%, SPL is 89.4%, and the correction rate is 12.7%. Three failure cases are highlighted in Fig. 8(b). The primary cause was inaccurate predictions, which led to a delayed obstacle avoidance. dditionally, when the drone performs an in-place rotation, minor lateral drift will increase the risk of side collisions.

D. *Real-life Experiments*

1) Hardware Configuration: We conduct experiments on a drone equipped with an onboard computer and a RealSense D435i camera. The RGB images have a resolution of 848×480 with a $69° \times 42°$ FoV, while the depth images share the same resolution but feature an $87° \times 58°$ FoV. Visual-Inertial Odometry (VIO) is executed onboard, but only the estimated linear velocity and attitude are used for reference tracking control in the PX4 autopilot. The live image stream is transmitted via 2.4 GHz Wi-Fi to an RTX 3060 laptop, with a latency of approximately 10 ms. The velocity control commands $u = [v_{lin}, v_{ang}]$ are converted from body frame to global frame commands $u' = [v_x, v_y, v_{ang}]$ using the yaw angle estimated by VIO, and then sent to the flight controller. The planning loop runs at approximately 20 Hz.

In Fig. 9, real-time velocity commands and the predicted collision probability Q_c are displayed in the upper-left corner of RGB and depth images. The navigation model explores the environment and approaches areas visually similar to the goal image. Fig. 10 shows the flight trajectory along with the predicted collision probabilities at each waypoint.

VI. Discussions nd Conclusions

In this work, we present SIGN, a novel end-to-end RL framework for ImageNav tasks on drones, demonstrating successful sim-to-real transfer. By integrating a visual navigation policy with a safety module, SIGN enables image-goal navigation with reliable collision avoidance. The navigation performance is thoroughly evaluated on three Habitat benchmarks, while the safety module is extensively tested in the Gazebo simulator. Overall, SIGN provides a practical and effective solution for deploying ImageNav on drones.

Our real-world experiments reveal potential failure cases in maze-like environments with visually similar features. When the target image has never been observed, the navigation policy may become stuck in loops, repeatedly exploring the same region without progress. In future work, we plan to deploy SIGN onboard the Jetson Orin NX platform with a RealSense camera. In preliminary tests, SIGN took 1.1 GB of GPU memory and the inference frequency reached 30 Hz when VIO was running simultaneously. The CPU utilization was about 35% and the power consumption was 12W.

REFERENCES

[1] X. Fan, M. Lu, B. Xu, and P. Lu, "Flying in highly dynamic environments with end-to-end learning approach," *IEEE Robotics and utomation Letters*, 2025.

[2] Y. Hu, Y. Zhang, Y. Song, Y. Deng, F. Yu, L. Zhang, W. Lin, D. Zou, and W. Yu, "Seeing through pixel motion: learning obstacle avoidance from optical flow with one camera," *IEEE Robotics and utomation Letters*, 2025.

[3] . Bhattacharya, N. Rao, D. Parikh, P. Kunapuli, Y. Wu, Y. Tao, N. Matni, and V. Kumar, "Vision transformers for end-to-end vision-based quadrotor obstacle avoidance," in *2025 IEEE International Conference on Robotics and utomation (ICR)*, 2025, pp. 1–8.

[4] . Bhattacharya, M. Cannici, N. Rao, Y. Tao, V. Kumar, N. Matni, and D. Scaramuzza, "Monocular event-based vision for obstacle avoidance with a quadrotor," in *Conference on Robot Learning*, vol. 270, 2024, pp. 4826–4843.

[5] Z. Xu, X. Han, H. Shen, H. Jin, and K. Shimada, "Navrl: Learning safe flight in dynamic environments," *IEEE Robotics and utomation Letters*, 2025.

[6] . Loquercio, E. Kaufmann, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza, "Learning high-speed flight in the wild," *Science Robotics*, vol. 6, no. 59, p. eabg5810, 2021.

[7] Z. l-Halah, S. K. Ramakrishnan, and K. Grauman, "Zero experience required: Plug & play modular transfer learning for semantic visual navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 031–17 041.

[8] X. Sun, P. Chen, J. Fan, T. H. Li, J. Chen, and M. Tan, "Fgprompt: fine-grained goal prompting for image-goal navigation," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2023, pp. 12 054–12 073.

[9] K. Yadav, . Majumdar, R. Ramrakhya, N. Yokoyama, . Baevski, Z. Kira, O. Maksymets, and D. Batra, "OVRL-V2: simple state-of-art baseline for imagenav and objectnav," *arXiv preprint arXiv:2303.07798*, 2023.

[10] . Majumdar, G. ggarwal, B. Devnani, J. Hoffman, and D. Batra, "Zson: Zero-shot object-goal navigation using multimodal goal embeddings," in *dvances in Neural Information Processing Systems*, vol. 35, 2022, pp. 32 340–32 352.

[11] . Szot, . Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. Chaplot, O. Maksymets, *et al.*, "Habitat 2.0: training home assistants to rearrange their habitat," in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, 2021, pp. 251–266.

[12] F. Xia, . R. Zamir, Z. He, . Sax, J. Malik, and S. Savarese, "Gibson env: Real-world perception for embodied agents," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9068–9079.

[13] . Chang, . Dai, T. Funkhouser, M. Halber, M. Niebner, M. Savva, S. Song, . Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," in *IEEE International Conference on 3D Vision*, 2017, pp. 667–676.

[14] S. K. Ramakrishnan, . Gokaslan, E. Wijmans, O. Maksymets, . Clegg, J. Turner, E. Undersander, W. Galuba, . Westbury, . X. Chang, *et al.*, "Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai," *arXiv preprint arXiv:2109.08238*, 2021.

[15] J. Xing, . Romero, L. Bauersfeld, and D. Scaramuzza, "Bootstrapping reinforcement learning with imitation for vision-based agile flight," in *Conference on Robot Learning*, vol. 270, 2024, pp. 2542–2556.

[16] Y. Tao, E. Iceland, B. Li, E. Zwecher, U. Heinemann, . Cohen, . vni, O. Gal, . Barel, and V. Kumar, "Learning to explore indoor environments using autonomous micro aerial vehicles," in *2024 IEEE International Conference on Robotics and utomation (ICR)*, 2024, pp. 15 758–15 764.

[17] D. S. Chaplot, R. Salakhutdinov, . Gupta, and S. Gupta, "Neural topological slam for visual navigation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12 875–12 884.

[18] E. Wijmans, . Kadian, . Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra, "DD-PPO: learning near-perfect pointgoal navigators from 2.5 billion frames," in *International Conference on Learning Representations*, 2020.

[19] L. Mezghan, S. Sukhbaatar, T. Lavril, O. Maksymets, D. Batra, P. Bojanowski, and K. lahari, "Memory-augmented reinforcement learning for image-goal navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 3316–3323.

[20] O. Kwon, N. Kim, Y. Choi, H. Yoo, J. Park, and S. Oh, "Visual graph memory with unsupervised representation for visual navigation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 15 890–15 899.

[21] N. Kim, O. Kwon, H. Yoo, Y. Choi, J. Park, and S. Oh, "Topological semantic graph memory for image-goal navigation," in *Conference on Robot Learning*, vol. 205, 2023, pp. 393–402.

[22] M. Hahn, D. S. Chaplot, S. Tulsiani, M. Mukadam, J. M. Rehg, and . Gupta, "No rl, no simulation: Learning to navigate without navigating," in *dvances in Neural Information Processing Systems*, vol. 34, 2021, pp. 26 661–26 673.

[23] . Sridhar, D. Shah, C. Glossop, and S. Levine, "Nomad: Goal masked diffusion policies for navigation and exploration," in *2024 IEEE International Conference on Robotics and utomation (ICR).* IEEE, 2024, pp. 63–70.

[24] D. Shah, . Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "Vint: foundation model for visual navigation," in *Conference on Robot Learning*, vol. 229, 2023, pp. 711–733.

[25] D. Shah, . Sridhar, . Bhorkar, N. Hirose, and S. Levine, "GNM: general navigation model to drive any robot," in *IEEE International Conference on Robotics and utomation*, 2023, pp. 7226–7233.

[26] . 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.

[27] M. damkiewicz, T. Chen, . Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager, "Vision-only robot navigation in a neural radiance world," *IEEE Robotics and utomation Letters*, vol. 7, no. 2, pp. 4606–4613, 2022.

[28] M. dang, J. Low, O. Shorinwa, and M. Schwager, "Singer: n onboard generalist vision-language navigation policy for drones," *arXiv preprint arXiv:2509.18610*, 2025.

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

[30] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," *arXiv preprint arXiv:2301.04104*, 2023.

[31] S. Fujimoto, P. D'Oro, . Zhang, Y. Tian, and M. Rabbat, "Towards general-purpose model-free reinforcement learning," in *The Thirteenth International Conference on Learning Representations*, 2025.

[32] D. Yarats, R. Fergus, . Lazaric, and L. Pinto, "Mastering visual continuous control: Improved data-augmented reinforcement learning," in *The Tenth International Conference on Learning Representations*, 2022.

[33] L. Zhang, Q. Zhang, L. Shen, B. Yuan, X. Wang, and D. Tao, "Evaluating model-free reinforcement learning toward safety-critical tasks," in *Proceedings of the I conference on artificial intelligence*, vol. 37, no. 12, 2023, pp. 15 313–15 321.

[34] K. P. Wabersich and M. N. Zeilinger, " predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *utomatica*, vol. 129, p. 109597, 2021.

[35] L. Brunke, M. Greeff, . W. Hall, Z. Yuan, S. Zhou, J. Panerati, and . P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *nnual Review of Control, Robotics, and utonomous Systems*, vol. 5, no. 1, pp. 411–444, 2022.

[36] K. Xiao, S. Tan, G. Wang, X. n, X. Wang, and X. Wang, "Xtdrone: customizable multi-rotor uavs simulation platform," in *2020 4th International Conference on Robotics and utomation Sciences (ICR S)*, 2020, pp. 55–61.