Appendix

1.*Is the decision network in this work trained solely for a specific scenario? If not, how does the system perform in environments other than those used in simulation? Additionally, what is its generalization capability across different environments?*

AS_1

We must make it clear that the decision/policy network in this work is not trained solely for a specific scenario. To further demonstrate the generalization capability, we provide a thorough analysis from both methodological and experimental result, and reformulate QS_1 into the following two subproblems:

(1) Why does the proposed method possess generalization capability?

(2) How well does the proposed method generalize to different environments?

Here, we present detailed answer to the above two subproblems:

- **(1) Why does the proposed method possess generalization capability?**

The policy network utilizes environmental information to adaptively regulate flight aggressiveness. However, there often exists a discrepancy in the distribution of environmental information between the training and the deployment phase. Therefore, it is crucial to ensure the generalization capability of the policy network across different environments. In this section, we will discuss the generalization capability of our method based on the following characteristics:

(a) Combining learning and model-based planner: Based on reinforcement learning, conventional end-to-end policy [1] directly map observed information to control variables $u\left(t\right)$, and is required to simultaneously consider multiple constraints such as safety, smoothness, and agile adaptation. Consequently, this end-to-end framework poses significant challenges to policy learning, resulting in unstable generalization capability across different environments. **In contrast, our work decouples agile adaptation from multiple constraints by combining learning with a model-based planner**. Specifically, we employ a model-based planner [2] to generate the trajectory by solving a constrained optimization in the form of

$$\begin{aligned}
&\min_{u(t),T} \int_0^T J\left(u(t)\right) + \rho\left(T\right), \\
&s.t. \\
&u(t) \in \mathcal{F}, \forall t \in \left[0, T\right], \\
&\mathcal{G}\left(u(t)\right) \preceq \mathbf{0}, \forall t \in \left[0, T\right], \\
&\mathcal{H}\left(u(t)\right) = \mathbf{0}, \forall t \in \left[0, T\right],
\end{aligned} \tag{1}$$

where $u\left(t\right)$ is the planning quantities, $\mathcal{F}$ the collision free region, $\mathcal{G}\left(\cdot\right)$ the dynamic constraints imposed by the physical limitation of the vehicle, and $\mathcal{H}\left(\cdot\right)$ summarizes the equality relations between the planning quantities and the states.

Based on this model, the level of aggressiveness can be represented by an inequality in $\mathcal{G}\left(\cdot\right)$, such as $h\left(\mathbf{v}\right) \preceq \bar{v}$, where $\mathbf{v}$ is the speed of vehicle and the hyperparameter $\bar{v}$ is dynamically calculated by the policy network. **In conclusion, we train the policy network exclusively for agile adaptation, avoiding the burden on generalization capability imposed by other constraints**.

(b) Environmental observations: Conventional learning-based approaches [1], [3] usually take the depth image $\mathcal{I}_t$ as environmental observations. However, as shown in Fig. 1(a), the noise of the on-board camera leads to the deviation of the observation distribution between the simulation and the real world. In addition, the limited field of view of the on-board camera cannot achieve omnidirectional perception, resulting in a lack of sufficient observation for the depth image $\mathcal{I}_t$ at the current frame. Therefore, it may be detrimental to the generalization capability to directly take the depth image as the environmental observation.

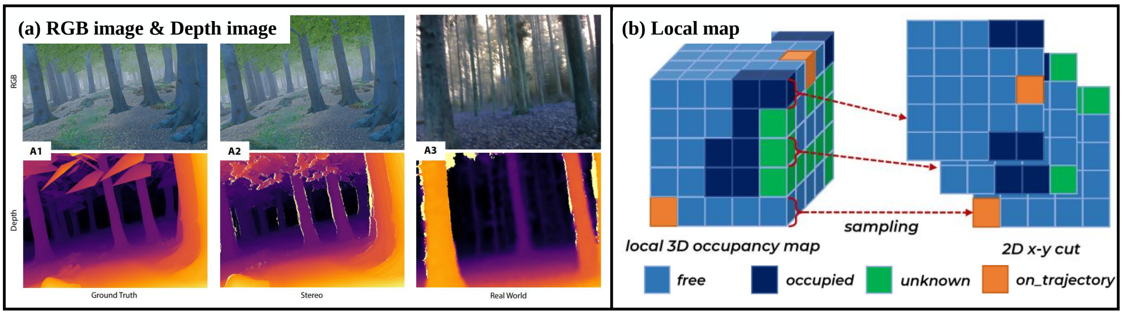

Figure 1: Illustration of environmental observation in different forms. **(a):** The effect of sensor noise on the depth image. For specific details, please refer to [3]. **(b):** The 3D occupancy map where each cell is assigned a state, and its x-y profiles are sampled as the input of the policy network [2].

In the state-of-the-art research [2], local maps $\mathcal{M}_t$ have been developed to represent the observations of the environment. It not only deals with sensor noise through the model-based approach, but also covers the environmental observations from $\mathcal{I}_{t-\triangle t}$ to $\mathcal{I}_t$. Stated differently, it naturally bridges the gap between simulation and reality in reinforcement learning training. Furthermore, [2] exploits the x-y 2D cuts of the 3D map corresponding to the sampled points on the trajectory as input to the network and designs a CNN encoder to extract features, which is illustrated in Fig. 1(b). Nevertheless, we figure that the CNN encoder is relatively inefficient for policy learning, especially in the human-in-the-loop reinforcement learning paradigm.

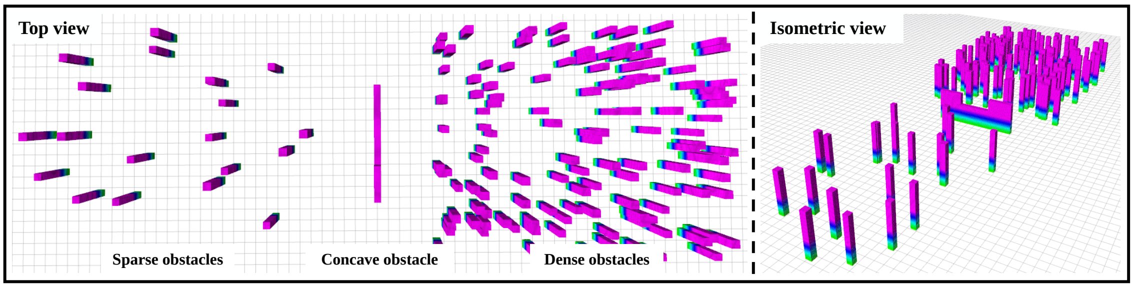

Figure 2: Screenshots of the training environment. Areas in colors represent the space filled with obstacles.

Our key insight is that the environmental observations represented by the 3D local map contain redundant information other than the spatial distribution of the trajectory and obstacles. Specifically, environmental features can be extracted manually in a more intuitive way instead of training a CNN encoder. For more efficient and easier learning, we sample the trajectory with time interval δt, keeping only the distances (calculated by Raycast) from the sampled points on the trajectory to the nearby obstacles in the 3D map as input to the network. **In summary, we manually extract the features regarding the spatial distribution of the trajectory and obstacles, avoiding the burden on the generalization capability brought by encoder training and sensor noise.**

(c) Reinforcement learning from human feedback: The aim of the policy network is to achieve safety-efficiency trade-off in navigation tasks. Specifically, the vehicle should adaptively regulate the flight aggressiveness when traversing environments with different levels of risk. To do this, reinforcement learning explores the action space through a trial-and-error mechanism and evaluates the action based on the reward function. While training policy via trial and error holds great promise, designing suitable reward functions remains challenging. For example, existing work evaluates environmental risk levels by measuring the distance from the vehicle to the nearby obstacles, or sparse collision rewards. However, such handcrafted reward functions still cannot effectively formulate the relationship between environmental observations and speed constraints.

In contrast, our work incorporates human feedback to fine-tune the policy network for generalization capability. Specifically, we first design handcrafted reward functions to achieve rough agile adaptation, and then combine human feedback to guide the policy network for generalization among different environments and pilots.

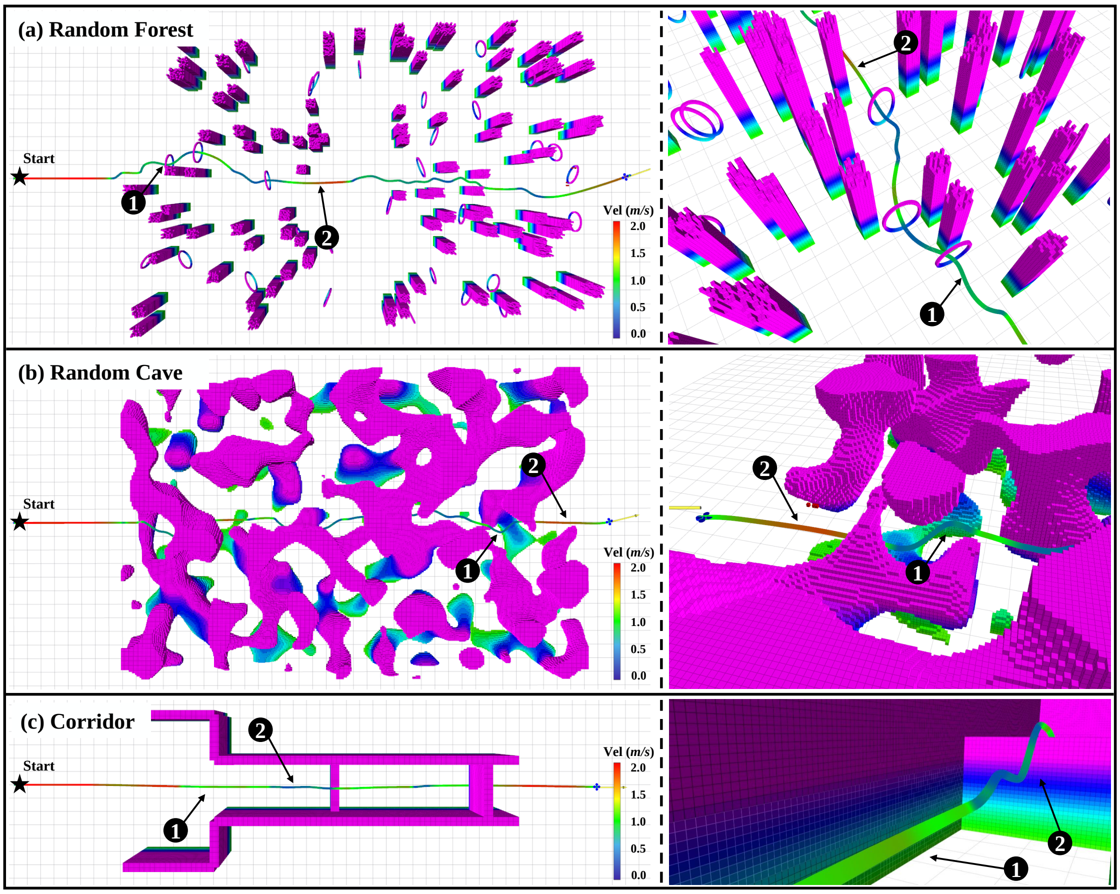

Figure 3: Experimental results on generalization capability in different environments. **Left:** Top views of different environments. **Right:** Close-ups of agile adaptation in the corresponding environments. The numbers ① and ② denote the corresponding relationship of positions in the environments.

- **(2) How well does the proposed method generalize to different environments?**

As discussed above, there is always a discrepancy in the distribution of environmental information across different environments. For example, obstacles with different densities or shapes. To introduce the generalization capability of the policy network, we have already discussed the superiority of the proposed approach from the perspective of the methodology. Subsequently, we describe the implementation details of our work in the training phase, and then present its experimental results across different environments.

(a) Training environment and implementation details: As shown in Fig. 2, the training environment has only cylindrical and concave obstacles. They are used to train the agile adaptation of the quadrotor in the x-y plane and the z-axis respectively. During the training phase, the quadrotor is asked to navigate along the reference path, receiving human feedback via joystick. Based on these, we train the policy network in the training environment with variable obstacle distributions, and the policy network is learned with the SAC algorithm [2]. As introduced in the manuscript, we first employ rewards with prior knowledge to guide the policy network, and then incorporate human feedback to fine-tune it. Note that we design a simulated human feedback to approximate real-world human-computer interaction. It generates the desired joystick input velocity by evaluating the occlusion of the field of view. For example, slight and severe occlusions correspond to medium (1.0m/s) and low (0.5m/s) human-desired speeds respectively.

(b) Test environments and experimental results: We design three test environments with various styles, which is illustrated in Fig. 3. For the random forest, we add new ring-shaped obstacles to evaluate the generalization capability. As shown on the right of Fig. 3(a), the policy network proactively reduced the flight speed to avoid the potential collision risk with the narrow ring-shaped obstacles. Furthermore, we verified the agile adaptation of the proposed policy network in the simulated random cave, as shown in Fig. 3(b). It must be emphasized that the random caves have completely different obstacle distribution characteristics compared with the training environment. Similarly, we design a simulated corridor that the policy network is also unfamiliar with. The experimental results prove that the policy network can still regulate the agile adaptation of the planner according to the width of the corridor. Particularly, we must emphasize that pilots tend to adopt a more conservative flight speed in altitude compared to that in the horizontal plane, as shown in the right of Fig. 3(c). In summary, the experimental results in multiple scenarios prove that the policy network in this work has significant generalization capability rather than being trained solely for a specific scenario.

2. *In the simulation study, do all three methods were presented to the subjects in the same order? If so, the subjects will have better idea of the environment, which gives advantages to the method used later.*

AS_2

In this appendix, we provide a thorough analysis of the experimental order and describe the corresponding solutions. We separate QS_2 into the following two subproblems and answer them accordingly for more clarification.

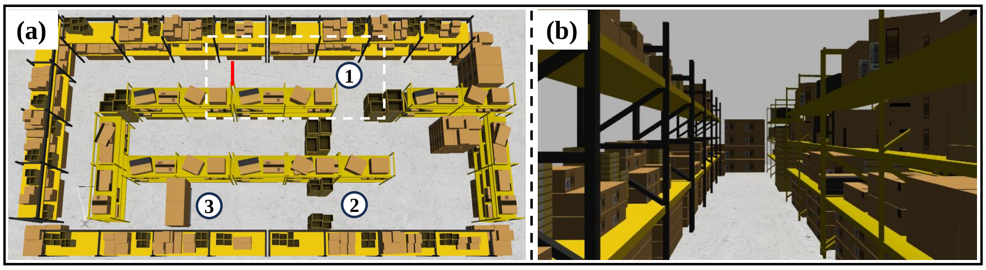

Figure 4: Screenshots of the simulated warehouse. **(a):** Subjects are tasked with teleoperating the quadrotor to patrol the warehouse, passing through ①, ②, and ③ in sequence. **(b):** The close-up of the environmental information within the white dotted box. The red line denotes the position of the quadrotor when the on-board camera first perceives the intersection.

- **(1) What is the effect of the experimental order on the benchmark comparison?**

In human-in-the-loop navigation, subjects make decisions based on environmental observations. For example, the quadrotor executes the planned trajectory of the previous human decision to patrol the environment. When the subject observes the intersection ahead, he/she may guide the quadrotor to perform replanning via the joystick, as illustrated in Fig. 4. However, we found two crucial phenomena during the teleoperation experiments. **(a)** Constrained by the urgency of the task (*e.g.*, search and rescue), subjects are required to complete the task as quickly as possible for the sake of navigation efficiency. **(b)** Due to the complex textures in the environment, subjects usually cannot identify intersections in advance and make informed decisions, which is illustrated in Fig. 4(b). Therefore, it is extremely challenging for subjects to guide the quadrotor to perform replanning in advance while maintaining navigation efficiency. To our knowledge, no existing work has taken into account the reaction time of subjects when making decisions at intersections. To relieve the cognitive burden of the subjects, we propose the HEA, which exploits the semantic information in the environment to adaptively regulate the flight speed of the quadrotor. To further verify the effectiveness of the proposed HEA, we designed simulation experiments to make comparisons with No-EA and EA.

As mentioned above, if all three methods were presented to the subjects in the same order, it will bring more advantages to the methods used later because the subjects have more prior knowledge of the environment.

To further quantify the impact of the prior information, we enlisted 12 subjects with varying proficiency levels to teleoperate the quadrotor through intersection ① in Fig. 4. We record the distance between the quadrotor and the red line when subjects observe the intersection and make decisions. Each subject teleoperates the quadrotor in eight trials.

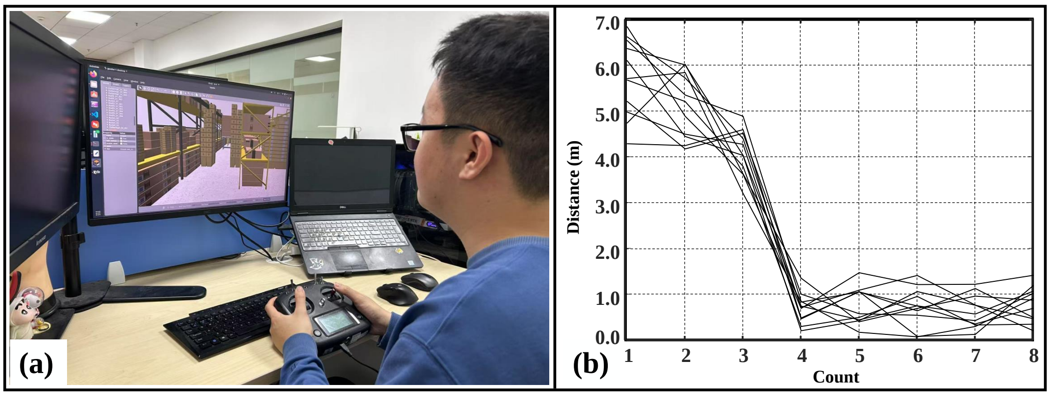

Figure 5: Experiments on the effect of prior information. **(a):** Illustration of subjects teleoperating the quadrotor in the simulation. **(b):** The plot of relevant data on the experimental rounds and the subjects' reactions.

As shown in Fig. 5, the results show that the subjects generally couldn't make decisions in advance when they teleoperated the quadrotor through the intersection for the first time, since the environment was unknown to them and had complex textures. As the number of experimental rounds increased, the subjects gradually grasped the prior information about the environment. In particular, they were able to guide the quadrotor to replan in advance starting from the fourth round. **Therefore, we argue that it is an unfair comparison to perform the No-EA, EA and HEA methods in sequence under the same environment settings.**

- **(2) How to eliminate the effect of the experimental order?**

To eliminate the effect of prior information caused by the experimental order, we redesigned three different simulated warehouses, among which the intersection ① was arranged in different areas. In addition, the obstacles in the simulated warehouses have similar texture and shape characteristics for fair comparison, which is illustrated in Fig. 6.

In the simulated environment, the same subject tested the No-EA, EA and HEA methods in scenarios (a), (b) and (c) of Fig. 6, respectively. To avoid heavy and unnecessary workload, three novice subjects were arranged to test the three methods separately in the same real-world environment.

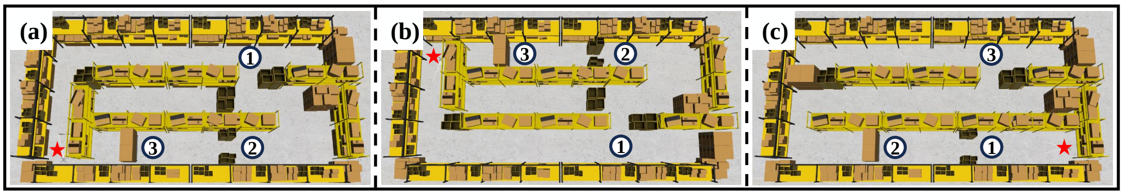

Figure 6: Screenshots of three simulated warehouses. The subjects are required to execute the No-EA, EA and HEA methods in the corresponding warehouses **(a)**, **(b)**, and **(c)**, respectively.

References

[1] Hang Yu, Christophe De Wagter, and Guido CH de Croon. “Mavrl: Learn to fly in cluttered environments with varying speed”. In: *arXiv preprint arXiv:2402.08381* (2024).

[2] Guangyu Zhao, Tianyue Wu, Yeke Chen, et al. “Learning Speed Adaptation for Flight in Clutter”. In: *IEEE Robotics and Automation Letters* (2024).

[3] Antonio Loquercio, Elia Kaufmann, René Ranftl, et al. “Learning high-speed flight in the wild”. In: *Science Robotics* 6.59 (2021), eabg5810.