drones

MDPI

Article

Imitation Learning of Complex Behaviors for Multiple Drones with Limited Vision

Yu Wan, Jun Tang * and Zipeng Zhao

College of System Engineer, National University of Defense Technology, Changsha 410003, China; wanyu13@nudt.edu.cn (Y.W.); zhaozipeng22@nudt.edu.cn (Z.Z.)
* Correspondence: tangjun06@nudt.edu.cn

Abstract: Navigating multiple drones autonomously in complex and unpredictable environments, such as forests, poses a significant challenge typically addressed by wireless communication for coordination. However, this approach falls short in situations with limited central control or blocked communications. Addressing this gap, our paper explores the learning of complex behaviors by multiple drones with limited vision. Drones in a swarm rely on onboard sensors, primarily forward-facing stereo cameras, for environmental perception and neighbor detection. They learn complex maneuvers through the imitation of a privileged expert system, which involves finding the optimal set of neural network parameters to enable the most effective mapping from sensory perception to control commands. The training process adopts the Dagger algorithm, employing the framework of centralized training with decentralized execution. Using this technique, drones rapidly learn complex behaviors, such as avoiding obstacles, coordinating movements, and navigating to specified targets, all in the absence of wireless communication. This paper details the construction of a distributed multi-UAV cooperative motion model under limited vision, emphasizing the autonomy of each drone in achieving coordinated flight and obstacle avoidance. Our methodological approach and experimental results validate the effectiveness of the proposed vision-based end-to-end controller, paving the way for more sophisticated applications of multi-UAV systems in intricate, real-world scenarios.

Keywords: perception and autonomy; multi-UAV system; sensor-based control; imitation learning; end-to-end controller

Citation: Wan, Y.; Tang, J.; Zhao, Z. Imitation Learning of Complex Behaviors for Multiple Drones with Limited Vision. *Drones* **2023**, *7*, 704. https://doi.org/10.3390/drones7120704

Academic Editor: Anastasios Dimou

Received: 23 October 2023
Revised: 23 November 2023
Accepted: 3 December 2023
Published: 13 December 2023

1. Introduction

The field of Unmanned Aerial Vehicles (UAVs) has seen substantial growth, with extensive studies conducted on efficient trajectory planners for single UAVs, resulting in numerous contributions [1–9]. However, decentralized planners for multi-UAV systems that can handle unknown, obstacle-dense environments remains an open problem. These planners are vital for controlling UAVs, facilitating conflict avoidance, task completion, and maintaining coordination and consistency at the group level. These complex systems encompass multiple interrelated functional aspects, such as communication and perception, formation and collision avoidance, and control and planning. The interplay and optimization of these components are crucial for the system's overall performance. Research on autonomous drone swarms, utilizing various navigation methods, including indoor motion capture [10–12], outdoor navigation using the Global Positioning System (GPS) [13–15], and vision-based [16–22], has provided significant insights. The current mainstream multi-UAV systems largely rely on wireless communication networks for information sharing, where UAVs share the location, speed, direction, and other data. This information forms the basis for group-coordinated actions, with each UAV planning its next move based on its relative position in the formation and with adjacent UAVs, thus maintaining a specific formation and avoiding individual conflicts. However, the use of wireless communication in multi-UAV systems presents certain drawbacks. The limited bandwidth can lead to network congestion and data transmission delays with an increase in the number of UAVs [18].

Furthermore, wireless communication can be susceptible to various forms of interference and attacks, resulting in data loss and errors. As an alternative, vision technology has emerged as a powerful supplement due to its large information capacity, good real-time performance, and resistance to interference.

Vision is indeed the primary sensory modality enabling collective motion in animal groups [20]. Drawing inspiration from natural biological swarms like bird flocks, vision has emerged as a potent sensory modality for agents. Due to its high information density, vision technology has the potential to significantly enhance the autonomy of UAVs. Onboard cameras present notable advantages over other UAV perception tools in terms of weight, cost, size, power consumption, and field of view. This trend is further supported by substantial advancements in computer vision and deep learning [23,24]. Vision inputs promptly detect shifts in the location or velocity of other drones within their field of view, bypassing the typical delays associated with wireless communication [18,19,23,24]. Moreover, vision offers environmental data, such as localization and obstacle information, aiding drones in executing complex maneuvers, including obstacle avoidance [7,8,25–27].

In the domain of multi-UAV systems, vision has been separately employed for perceiving neighboring entities and detecting obstacles, with no instances of its simultaneous application for both functions. When identifying neighboring entities, UAVs are typically equipped with omnidirectional cameras to capture global visual information [18,19,23,24]. This approach, while effective, imposes additional burdens in terms of weight and computational power. Intriguingly, certain studies have shown that bird flocks can attain superior coordination patterns under optimally limited fields of view than those under global fields of view [19,28]. This insight serves as an inspiration for our research on multi-UAV cooperation under limited vision.

The conventional control approaches for multi-UAV systems typically utilize either reactive or planning controllers. Reactive controllers generate commands according to a set of rules and integrate them, while planning controllers convert the control process into optimization problems, incorporating elements such as collision avoidance and formation maintenance as hard or soft constraints [8]. These methodologies represent hierarchical control strategies, deconstructing the navigation task into multiple subtasks. Each subtask is individually designed, debugged, and subsequently integrated with the rest.

Despite their ubiquity, these designs pose certain challenges. A key issue is the dependency on interrelated components, which requires each component to perform reliably and efficiently. Ignoring the interactions among different stages can result in compounded errors [29]. Additionally, this approach introduces extra delays that could hinder the execution of high-speed and flexible maneuvers [30]. The complex interdependencies within these hierarchical control systems often compromise the robustness against environmental shifts, variations in system parameters, or system failures.

These limitations have sparked increasing interest among researchers to investigate end-to-end strategies. These strategies aim to create a direct mapping from input to output, bypassing a sequence of intermediate states or processes [9,31–35]. Such an approach not only simplifies the complexity of the entire control system but may also improve performance and robustness by automatically adapting to various uncertainties and disturbances.

End-to-end strategies for multi-UAV systems often leverage machine learning or optimization techniques, including reinforcement learning methods such as sac [36], ppo [37], ddpg [38], etc., as well as imitation learning methods [9,33–35]. Among these, imitation learning holds particular promise for autonomous flight due to its unique advantages. It seeks to imitate the behaviors of experts or humans, thereby simplifying the learning process and optimizing actions based on the environment's state. This approach requires fewer sample complexities compared to reinforcement learning and can minimize the incidence of dangerous behaviors. Imitation learning can be trained through various methods, including human imitation [33], real-world experiences [34], or simulations [9,35]. Strategies that emulate model predictive control (MPC) controllers have successfully executed extreme acrobatic actions [9]. Additionally, end-to-end methods have facilitated

high-speed autonomous flight for quadrotors by simulating sample-based motion planning algorithms [35]. Notably, a fully visual end-to-end method can direct drone swarms using raw images, simulating clustering algorithms [39].

In this paper, we construct a distributed multi-UAV cooperative motion model for multi-UAV systems with limited vision, where each drone autonomously maneuvers, enabling coordinated flight and obstacle avoidance. Relying solely on onboard forward-facing cameras for neighbor detection and environmental perception, drones operate without wireless communication. We introduce an end-to-end controller built with neural networks to process visual inputs from grayscale and depth image streams, generating velocity control commands. Utilizing computer vision technology, specifically Yolo-V7 [40] and MobileNet [41] for aircraft detection and depth image feature extraction, we facilitate a compact representation for navigation in complex settings. The lightweight design of our policy network allows for high update rate onboard execution on quadrotors. The training involves an imitation learning approach within a centralized training with decentralized execution framework (CTDE), where an expert system provides demonstrations for optimal control. The training, inclusive of demonstration, data collection, and validation, is conducted in a Gazebo-simulated environment [42], iterating to refine the end-to-end perception-motion controller and minimize control signal discrepancies through the Dagger algorithm.

This paper's contributions can be summarized as follows:

1. We develop a motion model for UAVs with limited vision, where no wireless communication between UAVs is required. Each UAV relies solely on its local onboard camera to detect neighboring aircraft, sense environmental obstacles, and subsequently execute cooperative motion, navigating in unknown environments and avoiding obstacles.
2. We propose an end-to-end perceptual motion controller that directly implements sensor measurement to navigation command mapping, demonstrating exceptional generalization capabilities and robustness against perceptual artifacts, such as motion blur.
3. We apply imitation learning within the centralized training with decentralized execution (CTDE) frameworks to train the end-to-end controller. We construct a privileged expert system providing high-quality decision behavior data. The Dagger algorithm is adapted for the training process, facilitating the realization of visual input-to-control command mapping.
4. We adopt a decentralized control approach in the multi-UAV system, where individual agents rely exclusively on local data for collective decision making. Each agent is responsible for its decisions based purely on its own observations and local visual perception of nearby agents. This decentralization demonstrates superior robustness to single-point failures and exhibits strong scalability for large swarms.

The remainder of this paper is organized as follows: Section 2 details the proposed method. The effectiveness of the proposed method is verified by the simulation experiments in Section 3. Section 4 concludes the paper.

2. Method

In complex environments, each drone operates autonomously, eliminating the need for wireless communication. These drones rely primarily on onboard frontal binocular cameras, capable of detecting nearby drones and environmental obstacles, to navigate. Each drone utilizes an onboard end-to-end controller to control its movement. This controller guides the drone, modifying its flight path based on the detected environmental conditions and nearby entities. It adheres to a reference direction derived from a planning algorithm or user input, aimed at a target point, without directly considering potential conflicts or collisions.

Figure 1 displays the structure of our system. Our model is tripartite, grounded in the principles of imitation learning. It consists of an end-to-end controller, a privileged MPC controller, and a systematic training process. The end-to-end controller, built on a neural network architecture, serves as the 'student' in this context, aiming to learn from the 'teacher'—the MPC controller. As the teacher, the MPC controller leverages its access

to privileged information to provide high-quality demonstrations that guide the learning process of the end-to-end controller.

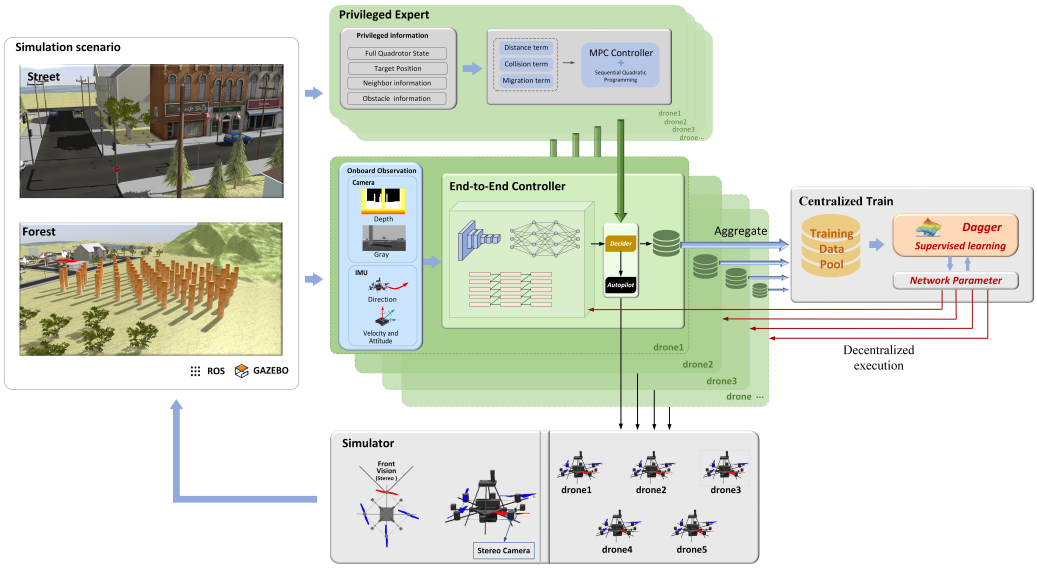


Figure 1. Overview of the proposed method: integration of an end-to-end controller, a privileged MPC controller, and a systematic training process.

The learning process revolves around identifying the optimal set of neural network parameters that enable an effective mapping from sensory perception to control commands. This process is facilitated through a method of centralized training with decentralized execution (CTDE), where the data from all the drones is compiled for the learning phase. We employ the Dagger algorithm, customized for this context, to train and fine-tune the neural network parameters. Once optimized, these parameters are individually deployed to each drone's end-to-end controller. This iterative process of learning and adaptation continues until the end-to-end controller achieves the desired proficiency in drone control.

2.1. End-to-End Controller

Each UAV is equipped with an onboard end-to-end controller responsible for sensing, cooperation, and motion planning. This controller incorporates a modular neural network, comprising a perception network module, a temporal convolution network module, and a policy generation network module. The controller accepts onboard sensor data as input and produces high-level control signals (speed commands) as output, thereby facilitating the mapping from the onboard sensor input to the high-level control command output. The end-to-end controller comprises three input branches, grayscale visual input, depth visual input, and data from inertial measurements (IMU), supplemented by the desired direction. Each of these inputs is processed by a corresponding perception backbone, followed by their integration into a temporal convolutional network and a multilayer perceptron for sequential action generation. This neural network framework is visualized in Figure 2.

The controller's input data encompass depth images $d \in \mathbb{R}^{540\times540}$ and grayscale images $d \in \mathbb{R}^{900\times450}$ obtained via the frontal binocular camera, the UAV velocity $v \in \mathbb{R}^3$, the acceleration $a \in \mathbb{R}^3$ and attitude data $q \in \mathbb{R}^9$ gathered by the IMU, and the intended flight direction $\omega \in \mathbb{R}^3$. Each type of input data is processed separately by the three perceptual branch networks. The resulting sequence of intermediate states is then fed into the temporal convolutional neural network to extract temporal features. These intermediate quantities are subsequently input into the control network for further processing. The final output is a set of advanced control signals (velocity commands) directed to the flight controller to manage the UAV's motion.

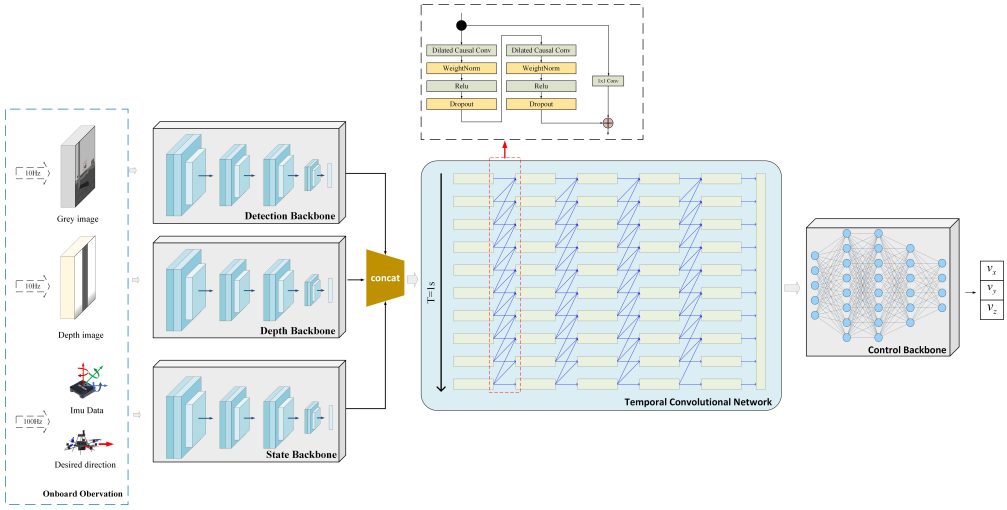
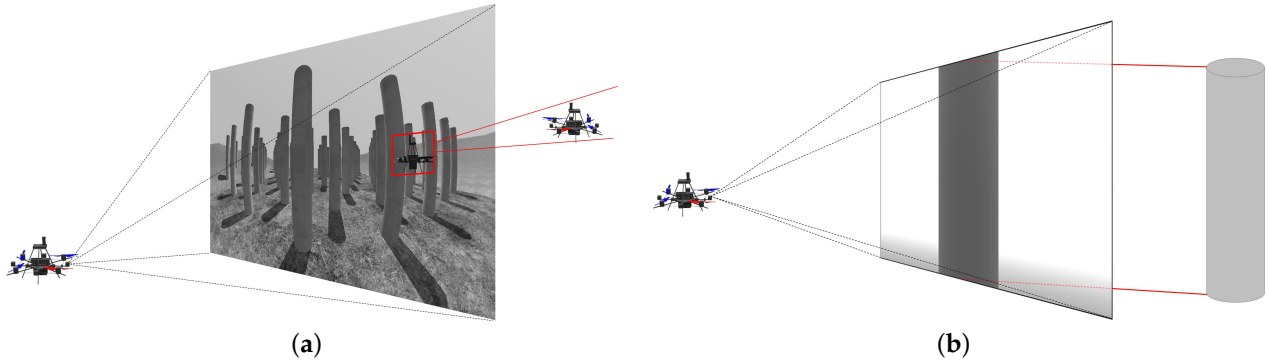


Figure 2. Network architecture of end-to-end controller. The visuomotor policy core takes camera observations, IMU data, and desired direction as inputs to generate commands.

In the first branch of the neural network, the input is sourced from grayscale image streams $d \in \mathbb{R}^{900\times450}$. These streams are produced by the front-facing camera of the UAV at a consistent sampling rate of 10 Hz, as illustrated in Figure 3a. The individual frames are channeled into an object detection layer, which employs the YOLOv7-tiny architecture. This architecture, having been pre-trained on our unique image dataset, is expertly fine-tuned to excel at the singular task of drone detection. The backbone of this system yields a four-dimensional feature vector $[x, y, \text{size\_}x, \text{size\_}y]$ for each drone identified in the image, encapsulating the object's spatial coordinates $[x, y]$ on the image, along with the dimensions of the detection box $[\text{size\_}x, \text{size\_}y]$. Subsequent to this detection process, the feature vector undergoes a dimension expansion by one tensor level, preparing it for input into a 2D convolutional neural network. This network comprises four hidden layers, each equipped with a distinct set of filters (32, 64, 128, 128), and is punctuated with LeakyReLU activation layers, fostering nonlinearity within the system. The processed data are then relayed through `globalAveragePooling2D`, which aids in reducing the spatial dimensions of the feature maps, while retaining their salient characteristics. The culmination of this branch sees the processed signals mapped onto a 128-dimensional feature vector via a fully connected layer, encapsulating a comprehensive representation of the identified drone's features in the image.



(**a**) (**b**)

Figure 3. Sample image from camera. (**a**) Sample image from a front grayscale camera. (**b**) Sample image from a front depth camera.

In the second branch of our neural network, we utilize a depth image stream $d \in \mathbb{R}^{540\times540}$ as the input. This stream is generated by the drone's front-facing stereo camera at a sampling frequency of 10 Hz, as depicted in Figure 3b. Each depth image frame undergoes

processing via a pre-existing MobileNet architecture, the purpose of which is to extract salient features from the depth image data. Subsequent to this extraction, the features are input into a 1D convolutional network. This network is composed of four hidden layers, each armed with filters (128, 64, 64, 64). Interleaved among these convolutional layers are LeakyReLU activation layers, designed to introduce nonlinearity and promote efficient learning. The final stage of this branch involves mapping the processed signals to a 128-dimensional feature vector. This transformation is achieved through a fully connected layer, bringing together the complex, multilayered information into a compact, yet richly informative, feature representation.

In the third branch of the neural network structure, the UAV's onboard inertial measurement unit (IMU) is leveraged, sampled at an impressive frequency of 100 Hz. This sampling encapsulates the UAV's current speed $v \in \mathbb{R}^3$, the attitude information represented by the rotation matrix $q \in \mathbb{R}^9$, and a reference motion direction $\omega \in \mathbb{R}^3$. It should be noted that the reference direction is a relatively crude estimate, which does not consider environmental influences. In the context of this research, it is defined as the unit direction vector extending from the UAV's present position toward the target point. This rich stream of information is processed through a five-layer perception network, fitted with filters (128, 64, 64, 64, 32), and interspersed with LeakyReLU activation layers. The culmination of this process sees the signals mapped onto a 128-dimensional feature vector, achieved via a fully connected layer.

The feature vectors outputted from the three diverse branches are concatenated and fed into a temporal convolution network, maintaining a time-stream of $T = 1$ s. This network employs a causal convolution, which is ideally suited for sequence models, facilitating the prediction of the state y_t using previously observed states $x_1, \ldots, x_t$. The fundamental unit of our network is a residual module known as TemporalBlock [43]. Each TemporalBlock comprises a dilated causal convolution layer, a norm layer, a ReLU layer, and a dropout layer. The dilated causal convolution layer, which is an enhanced convolution methodology featuring gaps, serves to increase the receptive field. This layer introduces a hyperparameter, the dilation rate, which we have set to 2. The entire temporal convolution network is arranged into five layers, a configuration that supports our aim of optimizing the processing and predictive capabilities of the temporal convolution network.

Subsequently, the inputs from each branch are integrated and processed through a multilayer perceptron. This perceptron encompasses four hidden layers, each equipped with filters (128, 64, 64, 64). The processed signals are then mapped onto a three-dimensional feature vector via a fully connected layer, signifying the desired speed $v \in \mathbb{R}^3$.

To maintain UAV stability and prevent abrupt movements from sudden speed changes, we impose a maximum speed limit $v_{\max}$. This is achieved by normalizing the network's output using a hyperbolic tangent (tanh) function to the range of $[-1, 1]$, which is subsequently scaled to meet $v_{\max}$. This method is commonly used for managing network outputs to ensure controlled and safe UAV speeds.

Additionally, we establish a maximum acceleration threshold $a_{\max}$ to regulate the changes in speed. We calculate the acceleration from the difference between the current command and the UAV's velocity state. If the acceleration's magnitude exceeds $a_{\max}$, it is normalized and scaled to this limit. After adjusting the acceleration, we then recompute the speed to ensure it aligns with these safe operational parameters. This process is essential for maintaining stable and safe acceleration, a critical factor given the UAV's field-of-view limitations and the necessity for controlled directional movement.

2.2. Privileged Expert

In our imitation learning framework, the end-to-end controller obtains high-quality decision-making behavior data from a privileged expert. Within a simulated environment, this expert is capable of 'cheating' by accessing privileged information, enhancing the demonstration quality. This includes access to its own ground-truth state (position $p \in \mathbb{R}^3$, velocity $v \in \mathbb{R}^3$, and acceleration $a \in \mathbb{R}^3$), as well as the ground-truth data of neighboring

agents and obstacle details ($p^{obs} \in \mathbb{R}^3$). A nonlinear model predictive control (NMPC) algorithm serves as this expert system.

The term $\mathrm{FOV}_i(t)$ is used to denote the field of view of agent i at time t. This field is visualized as a sector characterized by parameters ω (the angular width) and R (the range). The agent is situated at the center of this sector, with its symmetry axis aligning with the drone's forward direction, or line of sight (LOS). The sector's left and right boundaries form an angle of $\omega/2$ with the drone's LOS. During flight, the drone can acquire status information about neighboring drones within its field of view (FOV).

As indicated in Figure 4, $\mathcal{N}_i(t)$ signifies the set of neighboring drones for drone i at time t. To be included in this set, the neighbors must be within the drone's potential field and must not be obstructed by any obstacles. The neighbor set can be mathematically expressed as follows:

$$\begin{aligned}\mathcal{N}_i(t) &= \{j | p_j(t) \in \mathrm{FOV}_i(t), j = 1, \ldots, N\} \\ &= \left\{ j \middle| \|p_j(t) - p_i(t)\| \leq R \wedge \langle p_{ij}(t), L_i(t) \rangle \leq \omega/2 \wedge f(p_{ij}(t) - O_l) < 0, j = 1, \ldots, N \right\}\end{aligned} \tag{1}$$

Here, p_{ij} is the vector pointing from drone i to drone j, $L_i(t)$ represents the LOS direction of the drone i, $\langle p_{ij}, (L_i) \rangle$ signifies the angle between the two vectors, d_{ij}^2 denotes the Euclidean distance between drone i and drone j, and $f(\cdot)$ determines whether an obstacle blocks the line segment formed by the two positions.

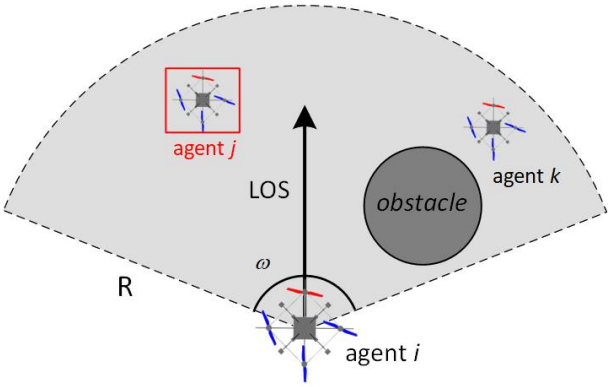


Figure 4. Illustration of agent potential fields: agent i considers agent j as a neighbor, and drone k is not deemed a neighbor to agent i due to obstacle interference.

During flight, the drone's LOS direction corresponds with its direction of motion, ensuring that the LOS is always directed toward the drone's path of motion. The drone can swing laterally during its flight in a random manner to expand its field of view while maintaining its primary focus forward, thereby dynamically extending its field of view to the sides. This pattern of movement increases the probability of neighboring drones entering its potential field. A random factor ε, which follows a normal distribution $\mathcal{G}(\mu, \sigma^2)$, is added to the LOS direction to further enhance the chances of neighboring drones entering the drone's field of view from the sides.

$$L_i(t) = \tan^{-1}\left(\frac{v_y}{v_x}\right) + \varepsilon \tag{2}$$

Planning for drones is conducted in a lower-dimensional planar output space, defined as $x = [p, v] = [p_x, p_y, p_z, v_x, v_y, v_z] \in \mathbb{R}^6$, which includes the drone position p, velocity v, and line acceleration a as the control vector u. $X(t) \in \mathbb{R}^{6NP}$ is defined as sequence of predicted states $x(t+l|t)$ for a future time period $l \in \{1, \ldots, P\}$, and $U(t) \in \mathbb{R}^{3NP}$ is the sequence of predicted controls $u(t+l|t)$ for the future time period $l \in \{1, \ldots, P-1\}$.

Assuming the drones obey a second-order discrete linear system, the dynamics of the drone under the MPC control system can be expressed as [12]:

$$x(t+1) = A(t)x(t) + B(t)u(t) + C(t)u(t-1) \tag{3}$$

Here, A, B, and C are constant matrices. x and u are referred to the world coordinate system.

For the drone i at moment t, various terms are defined, such as the separation term $J_{\text{sep},i}(t)$, the migration term $J_{\text{mig},i}(t)$, and the control term $J_{u,i}(t)$.

The separation term for drone i at moment t is defined as follows:

$$J_{\text{sep},i}(t) = \sum_{l=0}^{P} \sum_{j \in \mathcal{N}_i} \frac{w_{\text{sep}}}{|\mathcal{N}_i|} (\|p_j(t+l|t) - p_i(t+l|t)\|^2 - d_{\text{ref}}^2)^2 \tag{4}$$

This term combines the effects of aggregation and repulsion forces, allowing the drone to maintain a suitable distance d_{ref} with neighboring drones, where $j \in \mathcal{N}_i$ denotes the neighboring agents of drone i, and $\mathcal{N}_i$ represents the number of neighbors.

The migration term for drone i at moment t is defined as follows:

$$J_{\text{mig},i}(t) = \sum_{l=1}^{P} w_{\text{mig}} \left(\|v(t+l|t) - v^*(t+l|t)\|^2 \right) \tag{5}$$

$$v^*(t+l|t) = v_{\text{ref}} \left(\frac{goal - p(t+l|t)}{|goal - p(t+l|t)|} \right) \tag{6}$$

This term combines the effects of direction and velocity magnitude to move the drone toward the target point at the desired velocity. Here, the velocity direction only encodes the long-term target of the UAV, not its collision-free path.

The control term is defined as follows.

$$J_{u,i}(t) = \sum_{l=1}^{P-1} w_u \left(\|u(t+l|t)\|^2 \right) \tag{7}$$

In this case, $w_{\text{sep}}, w_{\text{mig}}, w_u$ are the constant weights associated with the cost function terms.

To prevent collisions between UAVs and their neighbors or obstacles, the cost function incorporates two sets of constraints:

$$d_{ij}(t+1|t)^2 \geq d_{\text{agent-safety}}^2, j \in N_i \tag{8}$$

$$d_{im}(t+l|t)^2 \geq d_{\text{obs-safety}}^2, m \in \{1, \ldots, M\} \tag{9}$$

Here, $d_{\text{agent-safety}}$ is the safety distance between two UAVs and $d_{\text{obs-safety}}$ is the distance between the UAV and an obstacle.

In summary, the whole non-convex optimization problem for drone i can be expressed as follows:

$$\min_{X(k), U(k)} \left(J_{\text{sep},i}(t) + J_{\text{mig},i}(t) + J_{u,i}(t) \right) \tag{10}$$

$$
\begin{aligned}
\text{Subject to} \quad & x(t+l+1|t) = Ax(t+l|t) + Bu(t+l|t), \\
& x(t|t) = x(t), \\
& v_{\min} \leq v(t+l|t) \leq v_{\max}, \\
& u_{\min} \leq u(t+l|t) \leq u_{\max}, \\
& d_{ij}(t+1|t)^2 \geq d^2_{\text{agent-safety}}, \\
& d_{im}(t+l|t)^2 \geq d^2_{\text{obs-safety}}, \\
& l \in \{1, \ldots, P\},\ j \in N_i.
\end{aligned}
$$

The proposed NMPC controller operates at a sampling frequency of 10 Hz with a prediction horizon of 2 s, which is divided into 20 intervals of 0.1 s each. We employ Casadi [44] to solve the NMPC problem, utilizing its formulation for a Nonlinear Programming (NLP) task.

The NMPC optimizes the open-loop control problem over a receding horizon of 20 time steps. The control command utilized for the training process, $a^{exp}_{i,t} = \pi_{expert}(s_{i,t})$, is derived from the second velocity in this optimized sequence. This command provides high-quality demonstrations for the end-to-end controller's learning process, ensuring effective training and accurate emulation of the expert policy.

2.3. Training Process

During the training process, we employ a centralized training with decentralized execution (CTDE) strategy, combined with the Dagger (dataset aggregation) algorithm, to train the student policy (end-to-end controller). Our primary objective was to establish an effective mapping from sensory inputs to control commands, thereby emulating the actions of a privileged expert policy and optimizing the neural network parameters. CTDE operates in two phases: execution and training.

In the execution phase (online and decentralized), each agent independently executes the policy based on local observations. For example, in the kth flight of drone i at time t, the expert system generates control commands $a^{\text{exp}}_{i,t}$ from privileged information $s_{i,t}$, while the student policy, using real-world observations $o_{i,t}$, produces $a^{\text{std}}_{i,t}$. These data, consisting of observations $o_{i,t}$ and student-generated commands $a^{\text{std}}_{i,t}$, are compiled into a training dataset.

In the training phase (offline and centralized), we aggregate these data to train a unified neural network. The Adam optimization algorithm is used to minimize the action discrepancy between the expert and student policies. The student system aims to match the expert policy's performance, guided by the following loss function:

$$
\pi_\theta = \min_{\hat{\pi}} \mathbb{E}_{o,s\sim\rho(\pi)} \left[\| a^{exp}_{i,t} - \hat{\pi}(o_{i,t}) \| \right] \tag{11}
$$

To counteract the behavioral drift and enhance the state-space coverage during training, the Dagger strategy is implemented [45] (Algorithm 1). This iterative method enables agents to gather additional data under the current policy while retaining expert labels. These new data, combined with the original dataset, form an aggregated dataset, thus explaining the term dataset aggregation. The agent's policy is then retrained on this enlarged dataset, with the process repeated for numerous iterations.

Algorithm 1: Dagger for multi-UAV flocking control

Input: Expert policy π_{expert}, iterations N **Initialize:** Empty Dataset $D \leftarrow \emptyset$, policy π_θ
Obtain initial expert dataset D_{expert} from environment using π_{expert}
Set $D \leftarrow D_{\text{expert}}$ **for** *episode* $k = 1, 2, \ldots$ **do**
Train policy π_θ on dataset D using Adam algorithm by minimizing the loss in Equation (11)
Initialize an empty dataset D_i for the current iteration $D_i \leftarrow \emptyset$
for *step* $t = 1, 2, \ldots$ **do**
for *drone* $i = 1, 2, 3$ ***in parallel*** **do**
Obtain local observation $o_{i,t}$ and privileged information $s_{i,t}$
Determine student action $a_{i,t}^{\text{std}}$ by executing π_θ under $o_{i,t}$
Determine expert action $a_{i,t}^{\text{exp}}$ by executing π_{expert} under $s_{i,t}$
if $|a_{i,t}^{std} - a_{i,t}^{exp}| < \xi$ **then**
Execute action $a_{i,t}^{\text{std}}$
else
Execute action $a_{i,t}^{\text{exp}}$
end
Store state-action pair $(o_{i,t}, a_{i,t}^{\text{exp}})$ into D_i
Update next local observation $o_{i,t+1}$ and privileged information $s_{i,t+1}$
end
end
Aggregate datasets $D \leftarrow D \cup D_i$
end
Output: Trained policy network π_θ

The pseudocode represents the algorithm utilized for multi-UAV imitation learning. The algorithm is iterated for a defined number of cycles N, each involving a training phase and an execution phase. During the initialization, the expert policy is utilized to generate an initial dataset D_{expert}, which is then assigned to the main dataset D. The policy under consideration, denoted as π_θ, is initially empty. In each iteration, the policy π_θ is trained on the accumulated dataset D. For every time step (from $t = 0$ to T), each drone detects its local observations $o_{i,t}$ and receives privileged information $s_{i,t}$. It then uses π_θ to determine its action $a_{i,t}^{\text{std}}$ and the expert policy π_{expert} to define the reference action $a_{i,t}^{\text{exp}}$. A decision mechanism is put in place to decide between the student action and the expert action to avoid collision under immature student policy. If the difference between the two proposed actions is less than a predefined threshold ξ and the execution of the student action does not lead to a predicted collision, the student action is implemented. Otherwise, the expert action is chosen. Every selected action and corresponding state are then recorded and aggregated into the dataset D_i for the current iteration. After processing all time steps, the dataset D_i is merged into the main dataset D. The process is repeated for the defined number of iterations N. At the end of these cycles, the final, trained policy π_θ is obtained, which can be shared for the UAVs' collective learning and actions. This Dagger algorithm is noteworthy for its capacity to aggregate data from both student and expert policies, helping the learning model to effectively generalize and improve over multiple iterations.

3. Experiments and Results

In this section, we meticulously evaluate the effectiveness of our proposed method concerning formation control and obstacle avoidance during multi-UAV operations, leveraging the Gazebo simulation environment. This robust platform renders a realistic simulation milieu mirroring the physical attributes of real-world robotics. We delve into the assessment of the learned vision-based swarm controller, juxtaposing its performance against a privileged expert controller and selected benchmark algorithms. This comparative insight

underscores various dimensions of our proposed system solution's performance potential. Moreover, we furnish quantitative assessments across several pertinent metrics for each application scenario.

To verify the proposed algorithm's functionality, we devised a simulated cityscape within the Gazebo environment (as shown in Figure 5). This simulation presents two complex scenarios for evaluation: (1) collective navigation in a street environment; (2) collective navigation through a forest. The Gazebo simulator, when used in synergy with the Betaflight firmware, provides state estimation and control, while the Robot Operating System (ROS) facilitates node communication. All the simulations are carried out within the Gazebo environment, with the trajectories displayed in Rviz, a three-dimensional visualization tool commonly employed in robotics research. Our experimental swarm comprises five quadrotor drones. Each drone is equipped with two simulated cameras: a grayscale camera and a stereo camera. The grayscale camera records 720 $\times$ 540 grayscale images at a rate of 20 frames per second, while the stereo camera captures 224 $\times$ 224 depth images, also at a frame rate of 20 Hz. These cameras, in conjunction with an onboard end-to-end controller built on TensorFlow, act on decentralized and onboard controlling of the drone.

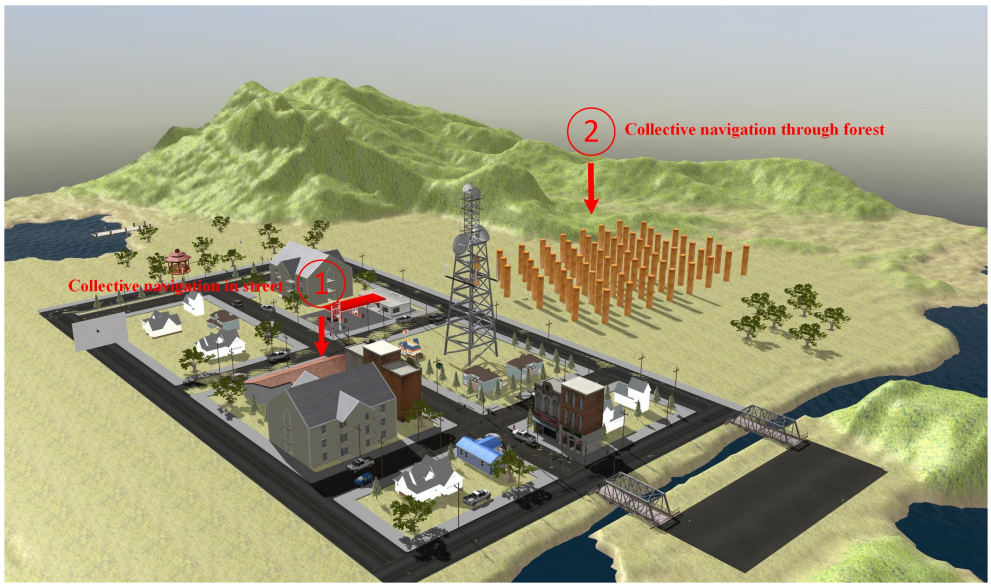


Figure 5. Illustration of the Gazebo simulation environment showcasing three scenarios: collective navigation in a street; formation navigation in a forest.

Our findings reveal that the proposed controller demonstrates considerable robustness, emerging as a viable alternative to communication-based systems, wherein the positional data of other agents are shared among the group members. This finding suggests the potential of our approach for enabling autonomous multi-UAV operations in complex real-world scenarios.

3.1. Collective Navigation in a Street

We explore a scenario wherein multi-UAVs are required to execute tasks parallel to real-world applications such as patrolling, courier services, and rescue operations, all maintaining their formation, avoiding inner-collision, and navigating along designated routes, akin to flying along a street (Figure 6). These applications are emblematic of tasks that demand a high level of coordination, precision, and adaptability.

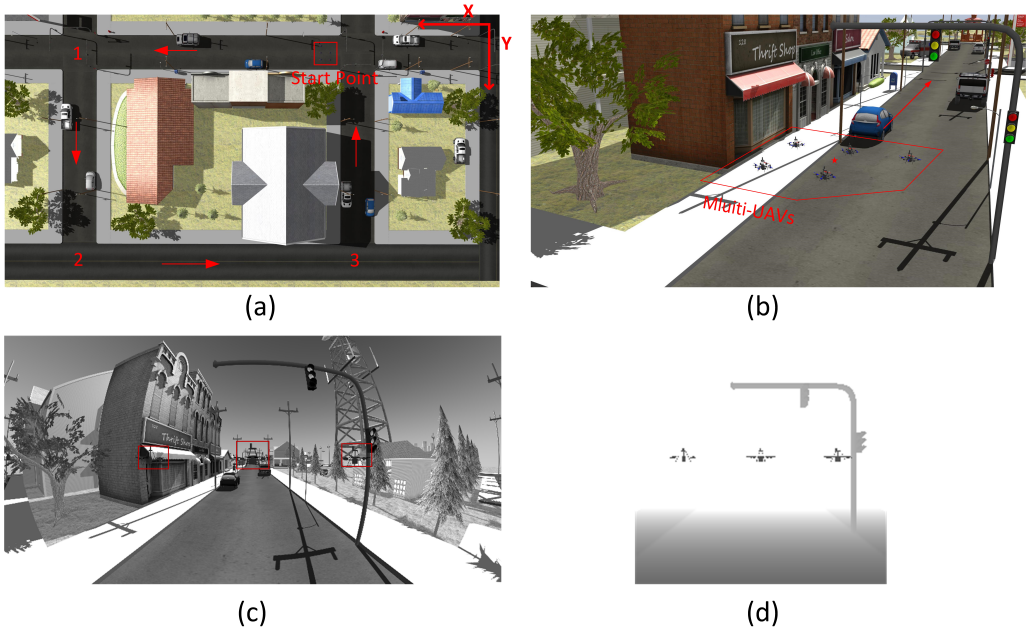


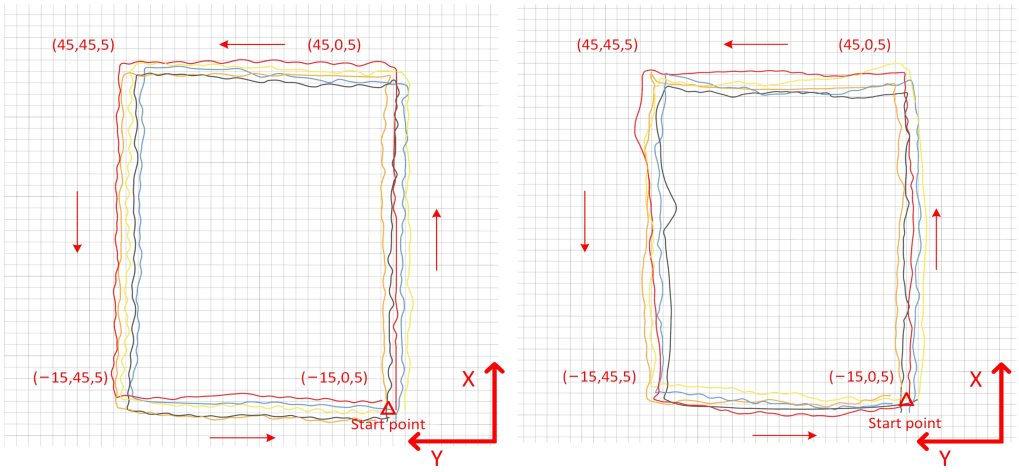
Figure 6. Detailed presentation of the Gazebo simulation environment: (**a**) aerial view of the Gazebo simulation space; (**b**) snapshot of multiple UAVs navigating a street; (**c**) real-time grayscale camera view from one of the participating drones; and (**d**) real-time depth camera perspective from one of the drones in flight.

Prior to each experimental flight, the UAVs are positioned approximately 2.5 m apart from each other (refer to the 'Start Point' in Figure 6a). Subsequently, all the drones are instructed to take off simultaneously and ascend to a prescribed altitude of 5 m. Once the drones reach the desired height, we engage the proposed onboard end-to-end controller that assumes decentralized control of their motion and orientation. This algorithm serves to replicate the collective and coordinated behavior observed in natural flocks, thus permitting the drones to maintain formation and navigate complex environments with minimal human intervention.

Each drone is programmed with an identical list of migration points to guide their path. These migration points function as interim destinations for the drones and ensure they follow a predetermined route. The algorithm switches to the next migration point as soon as an agent enters within an acceptance radius $r^{acc} = 3$ m from the current point, thereby ensuring continuous movement and a seamless navigation process.

Figure 6b illustrates a moment captured during the multi-drone flight. Figure 6c provides a real-time perspective from one of the drones in flight (marked by a pentagram in Figure 6). This perspective is captured using the YOLO-v7 algorithm, which identifies neighboring drones as indicated by the red bounding boxes. In contrast, the depth camera, as shown in Figure 6d, records the spatial relationships between the objects within its field of view, providing valuable data for assessing the drone's surrounding environment and for potential collision avoidance.

Figure 7a,b display the 2D aerial view of the multi-UAV trajectories as they navigate a street environment, controlled by expert and student policies, respectively, within the Rviz. By mimicking expert policy, the vision-only system successfully controls drones to sequentially navigate through three predetermined points, all the while avoiding internal collisions and preserving a cohesive formation.

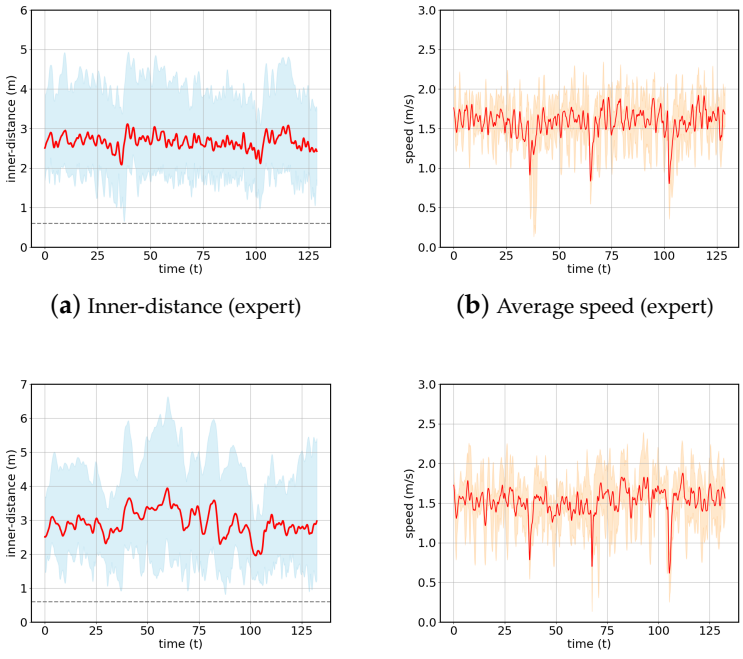


(**a**) Expert (**b**) Student

Figure 7. Two-dimensional aerial view of multiple UAV trajectories as they navigate through three target points in a street scenario: (**a**) controlled by a privileged expert policy utilizing the NMPC algorithm; and (**b**) guided by the proposed vision-limited policy.

These simulation results highlight the proposed vision-limited end-to-end controller's adaptability and efficiency in managing intricate scenarios. The centralized training with centralized execution, coupled with the Dagger algorithm, proves effective in enhancing the student system's ability to understand and replicate the expert system's control strategy.

Figure 8 provides a comprehensive comparative analysis between the expert and student policies in the street navigation scenario. Throughout the flight, both policies ensure that all agents avoid internal collisions, consistently maintaining a distance that exceeds the safety threshold.



(**a**) Inner-distance (expert) (**b**) Average speed (expert)

(**c**) Inner-distance (student) (**d**) Average speed (student)

Figure 8. Comparative analysis of simulation experiment outputs from privileged expert algorithm and vision-based policies: (**a**,**c**) inner-agent average distance (solid line) and range (shaded region), and (**b**,**d**) average speed (solid line) and range (shaded region).

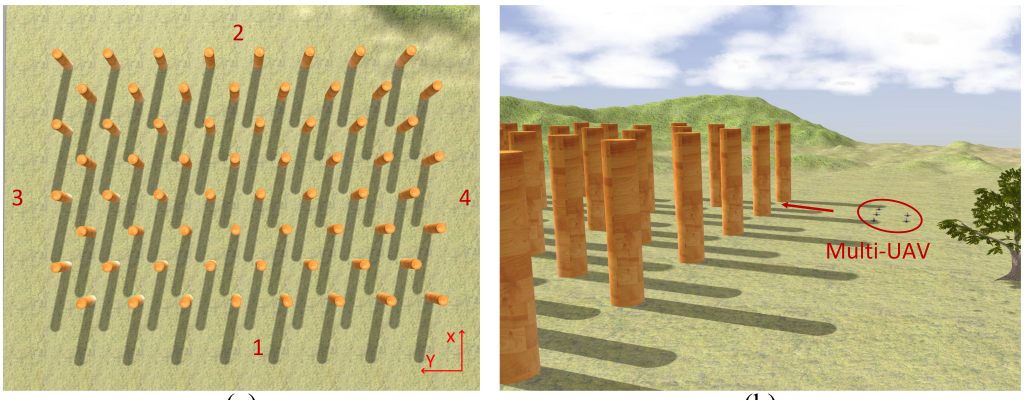
The performance of the expert, depicted in Figure 8a,b, establishes a standard for both stability and efficiency. Figure 8a shows the inner-agent distances to be stable, with relatively minor deviations from the mean, signifying a cohesive UAV formation within the structured street environment. The velocity profile of the expert policy, presented in Figure 8b, is marked by its uniformity, reflecting a smooth and synchronized movement with minimal velocity variations.

Correspondingly, the student policy outcomes, as displayed in Figure 8c,d, demonstrate a close alignment with the expert's performance, underlining the success of the imitation learning process. Figure 8c confirms that the student UAVs aptly replicate the expert's stable formation, keeping the inter-UAV distances within a secure and uniform range. Similarly, Figure 8d indicates that the student UAVs uphold a consistent average speed, paralleling the expert's steady velocity pattern. These observations underscore the student system's proficiency in navigating predictably and effectively in an environment with fewer complexities, such as urban streets.

3.2. Formation Navigation through Forest

We present an experiment designed to demonstrate fully autonomous swarm navigation in an environment with high obstacle density, mimicking a forest. The overarching aim is to enable the swarm of drones to navigate through the simulated forest while maintaining a consistent formation. This experiment is designed to demonstrate swarm navigation with full autonomy in a highly dense wild. The swarm should fly through the forests while staying in formation.

The simulated forest is constructed within a rectangular region, denoted as $R(l, w)$, with a width w of 75 m and a length l of 49 m. We populate this area with cylindrical obstacles, each with a radius of 1 m. These cylinders are methodically placed at regular intervals—7 m apart in the X direction and 10 m apart in the Y direction. As shown in Figure 9, this arrangement results in a highly dense simulated forest, emulating the challenging conditions of real-world woodland navigation.



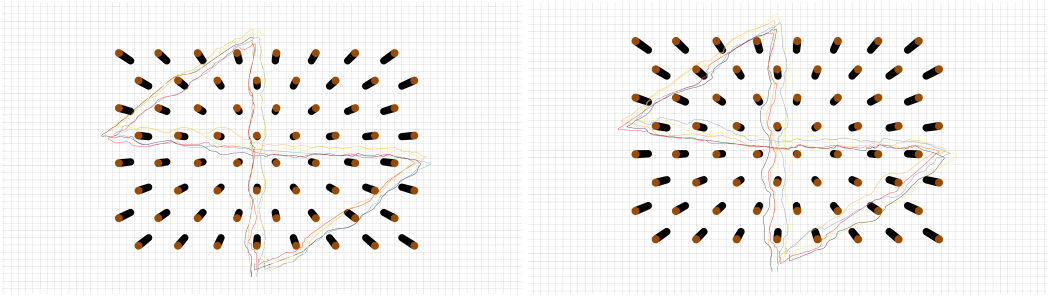
(a) (b)

Figure 9. Detailed illustration of the Gazebo simulation environment: (**a**) aerial view of the Gazebo simulation environment, populated with 49 uniformly distributed cylinders of 1m radius, spanning an area of roughly (49 m × 75 m); and (**b**) side view of the Gazebo simulation environment, exhibiting the drone formation after takeoff.

The swarm consists of five drones, which are initially arrayed in an orderly formation at the edge of the simulated forest (Figure 9b), with a lateral and longitudinal separation of 2 m between each drone. Throughout the course of the experiment, the swarm is tasked with reaching three designated target points in sequence and subsequently returning to the starting position. All the while, the drones must successfully avoid the obstacles and maintain an optimal formation to prevent dispersion.

Figure 10 illustrates the trajectories of the multi-UAVs navigating the forest under the control of expert (Figure 10a) and student policies (Figure 10b) within Rviz. Remarkably

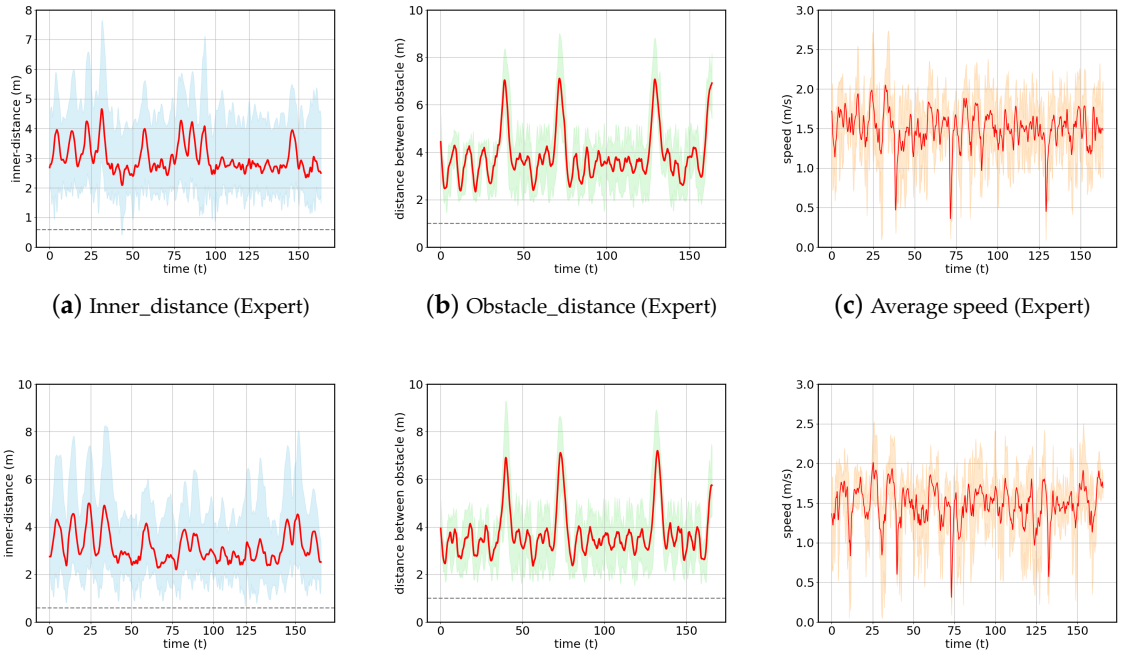
akin to the privileged expert system, the student policies successfully govern the drones, enabling sequential navigation through three target points, evading intra-collision and obstacles, and preserving a unified formation.



(**a**) 3D Aerial View (Expert) (**b**) 3D Aerial View (Student)

Figure 10. Visualization of multi-UAV trajectories navigating through a forest scenario in Rviz: (**a**) represents trajectories under the expert system (privileged NMPC algorithm), and (**b**) demonstrates trajectories guided by the student system (end-to-end sensorimotor controller).

In Figure 11, we present an analysis of both the expert and student policies maneuvering through a forest. The comparative performance between the expert (Figure 11a–c) and student (Figure 11d–f) systems indicates minimal discrepancies, showcasing the successful replication of the flight strategies. Notably, the inter-UAV distances during the flight, as shown in Figure 11a,d, consistently exceed the safety margins, ensuring collision avoidance throughout the experiment.



(**a**) Inner_distance (Expert) (**b**) Obstacle_distance (Expert) (**c**) Average speed (Expert)

(**d**) Inner_distance (Student) (**e**) Obstacle_distance (Student) (**f**) Average speed (Student)

Figure 11. Comparative analysis of simulation outputs between the expert (privileged NMPC controller) and student (end-to-end sensorimotor controller) policies: (**a**,**d**) inner-agent average separation (solid line) and dispersion (shaded region); (**b**,**e**) minimum distance between agents and obstacles (solid line) and range (shaded region); and (**c**,**f**) average speed (solid line) and variation (shaded region).

The expert policy, depicted in Figure 11a through Figure 11c, benchmarks the maneuverability within this complex terrain. Figure 11a illustrates the maintenance of stable

inter-drone distances, reflecting a controlled formation among the UAVs. Figure 11b further highlights the expert UAVs' proficiency in navigating around obstacles, maintaining a safe buffer that underscores their sophisticated obstacle-avoidance capabilities.

Mirroring these results, the student policy outcomes in Figure 11d–f demonstrate the policy's adaptability to the intricacies of the forest environment. Figure 11d shows the student UAVs' consistent adherence to formation protocols, while Figure 11e provides evidence of the successful emulation of the expert's obstacle-avoidance tactics, with the UAVs safely navigating around obstructions. The velocity trends, depicted in Figure 11f, reveal adaptive speed control, with the UAVs decelerating in dense obstacle regions and accelerating in open areas to maintain an optimal pace.

The analytical outcomes of Figure 11 attest to the student policy's ability to closely mimic the expert's advanced navigational strategies. The data demonstrate that even within the multifaceted forest environment, the student UAVs display a level of intelligent behavior that affirms the robustness of the imitation learning framework employed. This paves the way for future research into autonomous UAV operations in similarly complex and unpredictable settings.

3.3. Comparative Analysis

In order to compare the performance difference between our method and other benchmark algorithms, we define the performance metrics and conduct a relevant analysis. The benchmark algorithms are those such as sac [36], ppo [37], and ddpg [38]. We assessed the performance of the five policies in the pre-established scenario, ensuring that the environmental settings remained consistent with those during training. Each test comprised 20 episodes.

3.3.1. Performance Metrics

For a quantitative appraisal of our approach, we delineate the subsequent performance metrics:

1. Obstacle-Avoidance Index: This metric quantifies the nearest distance between the UAVs and any obstacle encountered during an episode, as expressed by:

$$d_{\min}^{obs} = \min(D^{obs}) \tag{12}$$

where

$$D^{obs} = \{d \,|\, d = \min_{k \in \forall obs} d_i^{obs}(t),\ i = 1, 2, \ldots, N,\ t = 1, \ldots, T\} \tag{13}$$

2. Conflict-Avoidance Index: This metric signifies the closest proximity between any two UAVs throughout an episode:

$$d_{\min}^{uav} = \min(D^{uav}) \tag{14}$$

where

$$D^{uav} = \{d \,|\, d = d_{ij}(t),\ i, j = 1, 2, \ldots, N,\ t = 1, \ldots, T\} \tag{15}$$

and $d_{ij}(t)$ is the distance between UAV i and UAV j at time t.

3. Group Index: This metric epitomizes the average distance among the UAVs throughout an episode:

$$d_g = \frac{1}{T}\frac{1}{N}\frac{1}{|\mathcal{N}_i|}\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j \in \mathcal{N}_i} d_{ij}(t) \tag{16}$$

$\mathcal{N}_i$ represents the neighbor set of aircraft i.

4. Order Index: This metric evaluates the consistency in alignment within the formation. A score of 1 denotes impeccable alignment, while a score of 0 indicates opposing directions:

$$\phi_{\text{order}} = \frac{1}{T}\frac{1}{N}\frac{1}{|\mathcal{N}_i|}\sum_{t=0}^{T}\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i}\frac{v_i(t)\cdot v_j(t)}{\|v_i(t)\|\|v_j(t)\|} \tag{17}$$

5. Average Speed: This metric represents the mean velocity of the UAVs throughout an episode:

$$V = \frac{1}{T}\frac{1}{N}\sum_{t=1}^{T}\sum_{i=1}^{N} v_i(t) \tag{18}$$

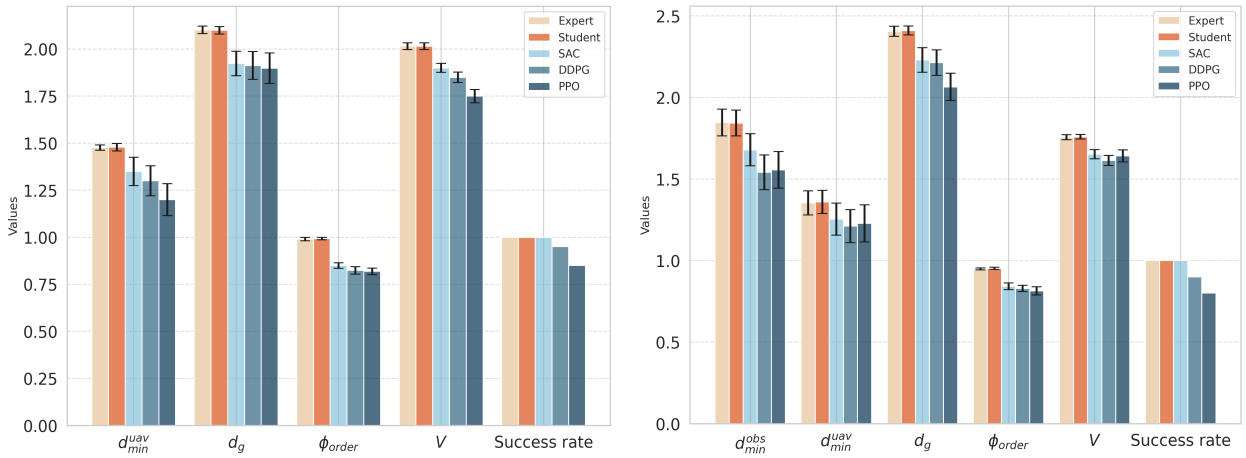
3.3.2. Performance Analysis

The analysis of the data, as presented in Table 1 and illustrated in Figure 12, reveals a comparative evaluation of the average and standard deviation of the performance metrics across different methods. A noteworthy observation from the analysis is that our proposed method significantly surpasses the performance metrics of the other three benchmark algorithms (sac [36], ppo [37], and ddpg [38]), highlighting the superior efficacy of our method over alternative strategies. This superior performance is not merely confined to a comparison with benchmark algorithms, but it extends to a close alignment with the metrics directed by the expert policy, underscoring the potent capability of imitation learning within this framework.

In this setting, the expert, equipped with privileged information, demonstrates behaviors that are mimicked by learners using local vision. This mechanism empowers the learners with a nuanced comprehension of their environment, enabling them to adeptly navigate through it. Particularly, our agent proficiently leveraged the training dataset furnished by the expert, crafting a strategy that endeavors to mirror expert-like behaviors with a high degree of fidelity. This embodiment of imitation is pivotal as it bridges the knowledge gap, allowing the student policy to closely emulate the expert policy's performance metrics. Such a replication is indicative of not just the method's competency but its potential for robust application in more complex or dynamic scenarios. Through a meticulous analysis and comparison, the efficacy and potential superiority of our method in navigating UAVs autonomously in complex scenarios have been substantiated.

Table 1. Comparative analysis: our method versus state-of-the-art baseline algorithms.

Scenario	Method	$d_{\min}^{obs}$(m)	$d_{\min}^{uav}$(m)	d_g(m)	ϕ_{order}	V(m/s)	Success Rate
Street	Expert	\\	1.477 ± 0.014	2.102 ± 0.020	0.991 ± 0.009	2.016 ± 0.018	1
	Student	\\	1.479 ± 0.020	2.100 ± 0.020	0.993 ± 0.007	2.015 ± 0.018	1
	SAC	\\	1.350 ± 0.075	1.924 ± 0.065	0.850 ± 0.015	1.900 ± 0.024	1
	DDPG	\\	1.300 ± 0.080	1.913 ± 0.074	0.824 ± 0.020	1.850 ± 0.028	0.95
	PPO	\\	1.200 ± 0.085	1.898 ± 0.081	0.819 ± 0.017	1.750 ± 0.035	0.85
Forest	Expert	1.847 ± 0.082	1.354 ± 0.074	2.406 ± 0.031	0.949 ± 0.006	1.756 ± 0.016	1
	Student	1.843 ± 0.079	1.359 ± 0.071	2.411 ± 0.028	0.952 ± 0.007	1.759 ± 0.014	1
	SAC	1.679 ± 0.098	1.254 ± 0.098	2.230 ± 0.075	0.842 ± 0.021	1.652 ± 0.028	1
	DDPG	1.541 ± 0.106	1.211 ± 0.101	2.214 ± 0.078	0.829 ± 0.019	1.614 ± 0.031	0.90
	PPO	1.556 ± 0.112	1.228 ± 0.114	2.065 ± 0.084	0.814 ± 0.025	1.641 ± 0.037	0.80



(**a**) Street (**b**) Forest

Figure 12. Performance comparison of our method versus state-of-the-art baseline algorithms in street and forest environments: a bar chart analysis.

4. Conclusions

The remarkable progress made in the autonomous flight of multiple drones in intricate, uncharted settings, such as forests, is significantly notable. However, these advancements typically lean on wireless communication for coordination, presenting challenges in environments where communication may be hindered or unavailable. This paper has therefore embarked on a thorough investigation of the autonomous learning of complex behaviors by multiple drones under limited vision, devoid of central control and communication. In our research, we have pioneered a distributed motion model for multi-UAV cooperative movement under vision constraints. This model facilitates an autonomous control mechanism where each drone independently directs its own motion while achieving collective coordination and obstacle avoidance. This collective intelligent behavior relies solely on onboard forward-facing cameras, facilitating the detection and tracking of neighboring drones and sensing environmental obstacles. Additionally, this paper introduced an innovative end-to-end controller developed with neural networks, which effectively translates visual inputs into control commands. To enhance the performance of this controller, we capitalized on modern computer vision technology, utilizing Yolo-V7 for drone detection and MobileNet for depth image feature extraction. To ensure robust navigation in complex environments, we designed the policy network to be extremely lightweight, enabling onboard execution on the quadrotor at high frequencies. To train the end-to-end controller, we employed an imitation learning mechanism. In this structure, the privileged expert system provides optimal control commands based on comprehensive environmental representations and perfect quadrotor state knowledge. These data are then used to train the student system to produce similar control commands, thus achieving efficient and effective autonomous behavior. The approach presented in this paper not only offers a model for cooperative motion of UAVs under limited vision but also a robust end-to-end controller that directly maps sensor measurements to navigation commands. Moreover, our application of imitation learning proves effective and dynamically adaptable, revealing potential for 'intelligent' behavior. To summarize, the decentralized control approach we have adopted demonstrates superior robustness against single-point failures and strong scalability to large swarms, while the imitation learning mechanism displays dynamic adaptability. Therefore, this study provides an essential foundation for further exploration and development in the realm of multi-UAV autonomous flight within complex and unknown environments. Our future work will aim at validating these approaches in real-world scenarios and enhancing the controller's performance in increasingly complex environments.

Author Contributions: Conceptualization, Y.W., J.T. and Z.Z.; methodology, Y.W.; software, Y.W. and Z.Z.; validation, Y.W., J.T. and Z.Z.; formal analysis, Y.W., J.T. and Z.Z.; investigation, Y.W., J.T. and Z.Z.; resources, Y.W., J.T. and Z.Z.; data curation, Y.W., J.T. and Z.Z.; writing—original draft preparation, Y.W.; writing—review and editing, Y.W.; visualization, Y.W., J.T. and Z.Z.; supervision, J.T.; project administration, J.T.; funding acquisition, J.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 62073330.

Data Availability Statement: The data are unavailable due to privacy or ethical restrictions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lopez, B.T.; How, J.P. Aggressive 3-D collision avoidance for high-speed navigation. In Proceedings of the ICRA, Singapore, 29 May–3 June 2017; Volume 1, pp. 5759–5765.
2. Florence, P.R.; Carter, J.; Ware, J.; Tedrake, R. Nanomap: Fast, uncertainty-aware proximity queries with lazy search over local 3d data. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; IEEE: Piscataway Township, NJ, USA, 2018; Volume 2, pp. 7631–7638.
3. Florence, P.; Carter, J.; Tedrake, R. Integrated perception and control at high speed: Evaluating collision avoidance maneuvers without maps. In Proceedings of the Algorithmic Foundations of Robotics XII: Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics, San Francisco, CA, USA, 18–20 December 2016; Springer: Berlin/Heidelberg, Germany, 2020; Volume 3, pp. 304–319.
4. Zhou, B.; Pan, J.; Gao, F.; Shen, S. Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight. *IEEE Trans. Robot.* **2021**, *37*, 1992–2009. [CrossRef]
5. Bucki, N.; Lee, J.; Mueller, M.W. Rectangular pyramid partitioning using integrated depth sensors (rappids): A fast planner for multicopter navigation. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4626–4633. [CrossRef]
6. Zhou, X.; Zhu, J.; Zhou, H.; Xu, C.; Gao, F. Ego-swarm: A fully autonomous and decentralized quadrotor swarm system in cluttered environments. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; IEEE: Piscataway Township, NJ, USA, 2021; Volume 6, pp. 4101–4107.
7. Tordesillas, J.; How, J.P. MADER: Trajectory planner in multiagent and dynamic environments. *IEEE Trans. Robot.* **2021**, *38*, 463–476. [CrossRef]
8. Zhou, X.; Wen, X.; Wang, Z.; Gao, Y.; Li, H.; Wang, Q.; Yang, T.; Lu, H.; Cao, Y.; Xu, C.; et al. Swarm of micro flying robots in the wild. *Sci. Robot.* **2022**, *7*, eabm5954. [CrossRef] [PubMed]
9. Kaufmann, E.; Loquercio, A.; Ranftl, R.; Müller, M.; Koltun, V.; Scaramuzza, D. Deep drone acrobatics. *arXiv* **2020**, arXiv:2006.05768.
10. Kushleyev, A.; Mellinger, D.; Powers, C.; Kumar, V. Towards a swarm of agile micro quadrotors. *Auton. Robot.* **2013**, *35*, 287–300. [CrossRef]
11. Preiss, J.A.; Honig, W.; Sukhatme, G.S.; Ayanian, N. Crazyswarm: A large nano-quadcopter swarm. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; IEEE: Piscataway Township, NJ, USA, 2017; Volume 9, pp. 3299–3304.
12. Soria, E.; Schiano, F.; Floreano, D. Predictive control of aerial swarms in cluttered environments. *Nat. Mach. Intell.* **2021**, *3*, 545–554. [CrossRef]
13. Vásárhelyi, G.; Virágh, C.; Somorjai, G.; Tarcai, N.; Szörényi, T.; Nepusz, T.; Vicsek, T. Outdoor flocking and formation flight with autonomous aerial robots. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; IEEE: Piscataway Township, NJ, USA, 2014; Volume 11, pp. 3866–3873.
14. Braga, R.G.; Da Silva, R.C.; Ramos, A.C.; Mora-Camino, F. Collision avoidance based on reynolds rules: a case study using quadrotors. In Proceedings of the Information Technology-New Generations: 14th International Conference on Information Technology, Las Vegas, NV, USA, 10–12 April 2018; Springer: Berlin/Heidelberg, Germany, 2018; Volume 12, pp. 773–780.
15. Vásárhelyi, G.; Virágh, C.; Somorjai, G.; Nepusz, T.; Eiben, A.E.; Vicsek, T. Optimized flocking of autonomous drones in confined environments. *Sci. Robot.* **2018**, *3*, eaat3536. [CrossRef]
16. Quintero, S.A.; Collins, G.E.; Hespanha, J.P. Flocking with fixed-wing UAVs for distributed sensing: A stochastic optimal control approach. In Proceedings of the 2013 American Control Conference, Washington, DC, USA, 17–19 June 2013; IEEE: Piscataway Township, NJ, USA, 2013; Volume 14, pp. 2025–2031.
17. Sanchez-Lopez, J.L.; Pestana, J.; de la Puente, P.; Suarez-Fernandez, R.; Campoy, P. A system for the design and development of vision-based multi-robot quadrotor swarms. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; IEEE: Piscataway Township, NJ, USA, 2014; Volume 15, pp. 640–648.

18. Schilling, F.; Schiano, F.; Floreano, D. Vision-based drone flocking in outdoor environments. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2954–2961. [CrossRef]
19. Couzin, I.D.; Krause, J.; James, R.; Ruxton, G.D.; Franks, N.R. Collective memory and spatial sorting in animal groups. *J. Theor. Biol.* **2002**, *218*, 1–11. [CrossRef]
20. Strandburg-Peshkin, A.; Twomey, C.R.; Bode, N.W.; Kao, A.B.; Katz, Y.; Ioannou, C.C.; Rosenthal, S.B.; Torney, C.J.; Wu, H.S.; Levin, S.A.; et al. Visual sensory networks and effective information transfer in animal groups. *Curr. Biol.* **2013**, *23*, R709–R711. [CrossRef] [PubMed]
21. Hu, T.K.; Gama, F.; Wang, Z.; Ribeiro, A.; Sadler, B.M. Vgai: A vision-based decentralized controller learning framework for robot swarms. *arXiv* **2020**, arXiv:2002.02308.
22. Hu, T.K.; Gama, F.; Chen, T.; Wang, Z.; Ribeiro, A.; Sadler, B.M. VGAI: End-to-end learning of vision-based decentralized controllers for robot swarms. In Proceedings of the ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; IEEE: Piscataway Township, NJ, USA, 2021; Volume 20, pp. 4900–4904.
23. Wu, Z.; Suresh, K.; Narayanan, P.; Xu, H.; Kwon, H.; Wang, Z. Delving into robust object detection from unmanned aerial vehicles: A deep nuisance disentanglement approach. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; Volume 21, pp. 1201–1210.
24. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Ling, H.; Hu, Q.; Nie, Q.; Cheng, H.; Liu, C.; Liu, X.; et al. Visdrone-det2018: The vision meets drone object detection in image challenge results. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018; Volume 22.
25. Kanellakis, C.; Nikolakopoulos, G. Survey on computer vision for UAVs: Current developments and trends. *J. Intell. Robot. Syst.* **2017**, *87*, 141–168. [CrossRef]
26. Artieda, J.; Sebastian, J.M.; Campoy, P.; Correa, J.F.; Mondragón, I.F.; Martínez, C.; Olivares, M. Visual 3-d slam from uavs. *J. Intell. Robot. Syst.* **2009**, *55*, 299–321. [CrossRef]
27. Faessler, M.; Fontana, F.; Forster, C.; Mueggler, E.; Pizzoli, M.; Scaramuzza, D. Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle. *J. Field Robot.* **2016**, *33*, 431–450. [CrossRef]
28. Huth, A.; Wissel, C. The simulation of the movement of fish schools. *J. Theor. Biol.* **1992**, *156*, 365–385. [CrossRef]
29. Zhang, Z.; Scaramuzza, D. Perception-aware receding horizon navigation for MAVs. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; IEEE: Piscataway Township, NJ, USA, 2018; Volume 26, pp. 2534–2541.
30. Falanga, D.; Kim, S.; Scaramuzza, D. How fast is too fast? the role of perception latency in high-speed sense and avoid. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1884–1891. [CrossRef]
31. Loquercio, A.; Maqueda, A.I.; Del-Blanco, C.R.; Scaramuzza, D. Dronet: Learning to fly by driving. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1088–1095. [CrossRef]
32. Sadeghi, F.; Levine, S. Cad2rl: Real single-image flight without a single real image. *arXiv* **2016**, arXiv:1611.04201.
33. Ross, S.; Melik-Barkhudarov, N.; Shankar, K.S.; Wendel, A.; Dey, D.; Bagnell, J.A.; Hebert, M. Learning monocular reactive uav control in cluttered natural environments. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; IEEE: Piscataway Township, NJ, USA, 2013; Volume 30, pp. 1765–1772.
34. Gandhi, D.; Pinto, L.; Gupta, A. Learning to fly by crashing. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; IEEE: Piscataway Township, NJ, USA, 2017; Volume 31, pp. 3948–3955.
35. Loquercio, A.; Kaufmann, E.; Ranftl, R.; Müller, M.; Koltun, V.; Scaramuzza, D. Learning high-speed flight in the wild. *Sci. Robot.* **2021**, *6*, eabg5810. [CrossRef]
36. Bai, C.; Yan, P.; Piao, H.; Pan, W.; Guo, J. Learning-based multi-UAV flocking control with limited visual field and instinctive repulsion. *IEEE Trans. Cybern.* **2023**, 1–14. [CrossRef] [PubMed]
37. Fang, Z.; Jiang, D.; Huang, J.; Cheng, C.; Sha, Q.; He, B.; Li, G. Autonomous underwater vehicle formation control and obstacle avoidance using multi-agent generative adversarial imitation learning. *Ocean. Eng.* **2022**, *262*, 112182. [CrossRef]
38. Yan, C.; Wang, C.; Xiang, X.; Low, K.H.; Wang, X.; Xu, X.; Shen, L. Collision-Avoiding Flocking With Multiple Fixed-Wing UAVs in Obstacle-Cluttered Environments: A Task-Specific Curriculum-Based MADRL Approach. *IEEE Trans. Neural Networks Learn. Syst.* **2023**, 1–15. [CrossRef] [PubMed]
39. Schilling, F.; Lecoeur, J.; Schiano, F.; Floreano, D. Learning vision-based flight in drone swarms by imitation. *IEEE Robot. Autom. Lett.* **2019**, *4*, 4523–4530. [CrossRef]
40. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
41. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, South Korea, 27 October–2 November 2019; Volume 36, pp. 1314–1324.
42. Koenig, N.; Howard, A. Design and use paradigms for gazebo, an open-source multi-robot simulator. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), Sendai, Japan, 28 September–2 October 2004; IEEE: Piscataway Township, NJ, USA, 2004; Volume 3, pp. 2149–2154.

43. Zhang, K.; Yang, Z.; Başar, T. Multi-agent reinforcement learning: A selective overview of theories and algorithms. In *Handbook of Reinforcement Learning and Control*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 321–384.
44. Andersson, J.A.; Gillis, J.; Horn, G.; Rawlings, J.B.; Diehl, M. CasADi: A software framework for nonlinear optimization and optimal control. *Math. Program. Comput.* **2019**, *11*, 1–36. [CrossRef]
45. Ross, S.; Gordon, G.; Bagnell, D. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 627–635.