2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

FlightDiffusion: Revolutionizing Autonomous Drone Training with Diffusion Model Generating FPV Video

Valerii Serpiva*, Artem Lykov*, Faryal Batool, Vladislav Kozlovskiy,
Miguel Altamirano Cabrera and Dzmitry Tsetserukou

***Abstract*— We present FlightDiffusion, a diffusion-based framework for training autonomous drones from first-person view (FPV) video. The model generates FPV video sequences from a single frame and a text prompt, and derives corresponding state-action trajectories for task-conditioned navigation. FlightDiffusion leverages generative modeling to synthesize diverse FPV trajectories and corresponding state-action pairs, enabling scalable dataset generation without the high cost of real-world data collection. These datasets support not only the learning pipeline but also the training of autonomous navigation systems. Our evaluation shows that the generated trajectories are physically feasible and executable, with a mean positional error of 0.25 m (RMSE 0.28 m) and a mean orientation error of 0.19 rad (RMSE 0.24 rad). This approach establishes scalable dataset generation and supports reliable navigation performance. Results in simulated environments indicate stable trajectory planning and consistent behavior across varying conditions. An ANOVA revealed no statistically significant difference between performance in simulation and reality (F(1, 16) = 0.394, p = 0.541), with success rates of M = 0.628 (SD = 0.162) and M = 0.617 (SD = 0.177), respectively, indicating effective sim-to-real transfer. The generated datasets provide a useful resource for future UAV research. This work introduces diffusion-based video generation as a promising mechanism for coupling task-level reasoning with executable trajectory synthesis in aerial robotics.**

I. Introduction

Autonomous navigation for unmanned aerial vehicles (UAVs) has transformed applications ranging from search and rescue to autonomous drone racing [1], and rapid response in unstructured environments [2]. A key barrier to deploying these systems robustly in complex, dynamic settings is the challenge of developing navigation policies that can reason about unseen situations and adapt in real-time. Traditional data-driven methods, such as reinforcement learning (RL) [3] and imitation learning (IL) [4], have shown promise but are notoriously sample-inefficient. They require vast and diverse datasets of flight experience, which are logistically complex, expensive, and often dangerous to collect in the real-world. This reliance on large-scale data creates a significant bottleneck for innovation and deployment.

To address these challenges, the field is increasingly turning towards generative models. Recent works have demonstrated the power of diffusion models to learn complex quadrotor dynamics [5] and generate specialized aerobatic trajectories [6]. Concurrently, the integration of large language models (LLMs), visual-language action, and vision-language models (VLMs) is endowing drones with high-level semantic reasoning. These models enable UAVs to interpret natural language commands [7], [8], perform advanced path planning [9], adapt their behavior based on contextual understanding [10], [11], and incorporate diffusion-based video generation for UAV video recording [12]. This has led to more intelligent and human-responsive systems for specialized tasks like large-scale mission execution [13].

However, a fundamental gap remains between high-level semantic planning and low-level, physically grounded control. While VLMs can infer what actions should be performed, generating precise visuomotor commands for how to execute them remains data-intensive and difficult to scale. To address this, we introduce FlightDiffusion, a framework that bridges this gap by integrating high-level reasoning with diffusion-based first-person view (FPV) video generation and trajectory reconstruction. Given a single input frame and task description, the system generates realistic flight sequences, which are then converted into 3D trajectories and state-action representations via visual odometry. This approach enables

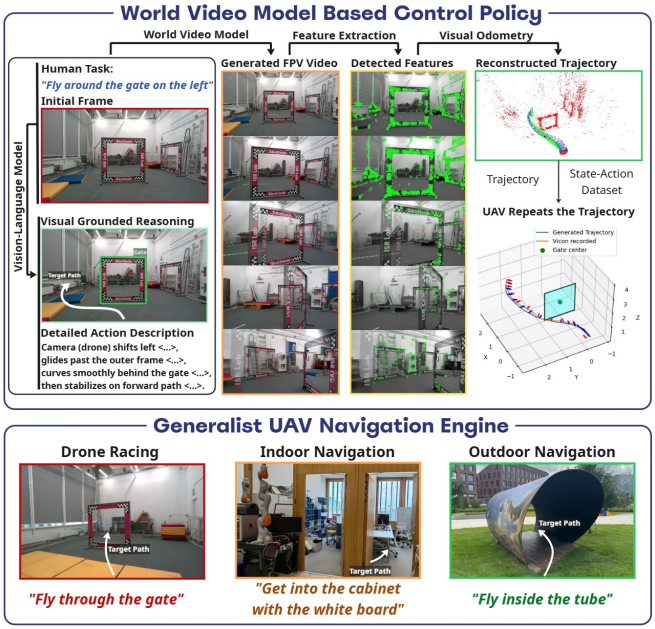

Fig. 1: FPV video generated from a single frame and task description using a diffusion model, followed by visual odometry for 3D trajectory and state-action reconstruction.

* These authors contributed equally to this work.

The authors are with the Intelligent Space Robotics Laboratory, Skolkovo Institute of Science and Technology Moscow, Bolshoy Boulevard 30, bld. 1, 121205, Moscow, Russia. {Valerii.Serpiva, Artem.Lykov, Faryal.Batool, Vladislav.Kozlovskiy, M.Altamirano, D.Tsetserukou}@skoltech.ru

a direct link between semantic reasoning and executable motion, supporting scalable training of navigation policies.

Our contributions are as follows:

1) We introduce FlightDiffusion, a framework that conditions image-to-video diffusion models on high-level task descriptions to generate physically consistent FPV flight videos for autonomous drone navigation.
2) We present a pipeline that converts generated FPV video into executable trajectory representations by reconstructing 3D motion and deriving UAV state-action pairs using visual odometry.
3) We show that FlightDiffusion functions as a scalable data generation pipeline, producing diverse, action-rich FPV datasets for UAV and robotics applications. This reduces reliance on costly real-world data collection and supports the development of scalable and diverse navigation datasets.

Unlike traditional approaches that depend on large-scale manual data collection, our framework leverages generative models to synthesize FPV video conditioned on high-level task descriptions, enabling scalable and diverse dataset generation with minimal human supervision. Instead of directly producing control commands, FlightDiffusion generates visually grounded flight sequences from which physically consistent trajectories and state-action pairs are reconstructed via visual odometry. This formulation bridges the gap between high-level semantic reasoning and UAV-control by transforming generative visual predictions into executable motion representations. As a result, the framework produces diverse, action-rich training data that improves policy robustness and reduces dependence on costly real-world data collection, providing a practical pathway toward more scalable autonomous navigation systems.

II. Related Work

A wide range of approaches has been explored, including control-theoretic methods, data-driven learning, and, more recently, generative and foundation models. Despite this progress, achieving robust generalization across tasks, environments, and embodiments remains an open problem. Existing methods often rely on large-scale real-world data or fail to tightly couple high-level reasoning with low-level control, motivating more scalable and integrated approaches.

A. Data-Driven UAV Navigation

Autonomous navigation for UAVs has long relied on data-driven methods such as RL and IL. RL enables policies to emerge from trial-and-error exploration, but it remains sample-inefficient and faces persistent sim-to-real transfer issues. IL offers a more direct route by using expert demonstrations [14], but its scalability is constrained by the expense of collecting large amounts of high-quality data. Recent advances attempt to overcome these barriers. [15] demonstrated RL-based UAV navigation in cluttered, GPS-denied spaces, showing adaptability but at the cost of intensive data requirements. Similarly, [16] proposed an IL framework for orchard environments that uses a variational autoencoder-based controller to map raw RGB inputs directly to controls, reducing reliance on explicit mapping but still requiring repeated expert interventions. These works underscore the need for scalable alternatives to raw experience collection.

B. Generative Models for UAV Dynamics

Generative models, particularly diffusion models, have recently emerged as a promising paradigm in aerial robotics. DroneDiffusion has been used to learn quadrotor dynamics directly from data, offering improvements over classical models [5], while [6] applied diffusion to synthesize aerobatic trajectories beyond the reach of conventional planners. Constraint-guided extensions such as Constraint-Guided Diffusion [17] further integrate safety and feasibility constraints, ensuring that the generated UAV trajectories remain dynamically valid and collision-free.

SwarmDiffusion [18] introduces an end-to-end diffusion model that jointly predicts visual traversability and generates feasible trajectories directly from RGB observations. PRESTO [19] introduces a learning-guided motion planning framework that generates seed trajectories using a diffusion model for trajectory optimization, enabling efficient, collision-free path generation in narrow-passage environments. GooDFlight [20] introduces a goal-oriented diffusion model for flight trajectory prediction, decoupling the process into goal estimation and trajectory prediction stages to improve accuracy, diversity, and interpretability. In a recent study, a conditional text-to-image diffusion model was proposed to generate a synthetic aerial dataset for UAV detection, demonstrating significant improvements in detection precision when trained on synthetic data and evaluated on real-world datasets [21].

A diffusion model has been introduced to characterize the flock generation process of UAV systems, with a proposed flocking control algorithm that effectively avoids collisions and analyzes system capacity through both simulation and real-world experiments [22].

C. Foundation Models for UAV Intelligence

The integration of foundation models has advanced UAVs toward semantic and contextual reasoning. VLMs and LLMs have been used to translate high-level instructions into executable goals [23], [9], [11]. NavFoM [24] introduced a unified foundation model trained on multiple robot embodiments and tasks, demonstrating robust generalization without task-specific tuning. VLM-RRT [25] guides sampling-based motion planning with semantic cues from language, demonstrating how VLMs can inject task awareness into low-level planners. Similarly, [26] compared synthetic and real training data for navigation, highlighting the persistent sim-to-real gap and the value of hybrid strategies. These works illustrate how foundation models can reason about *what* to do, but typically rely on hand-crafted or existing controllers for *how* to do it.

III. System Architecture

The FlightDiffusion system employs a modular pipeline comprising three core stages: Visual Reasoning and Mission Planning, Video Generation, and 3D Trajectory Reconstruction (Fig. 1), with the hardware configuration detailed in Table I.

The first module interprets the scene and generates high-level, semantically rich commands. We leverage Gemini 2.5 Flash [27], a powerful multimodal VLM from Google DeepMind, for this task. The pipeline is initiated by a single RGB image, denoted as I_{input}, which is captured from an onboard drone camera. This image is then converted into a standardized tensor, I_{std}, which is fed into the Gemini API alongside a structured natural language prompt engineered to elicit specific reasoning about the environment and the desired drone maneuver. This approach allows Gemini 2.5 Flash to perform long-horizon planning tasks implicitly by understanding complex scene geometry and objects (e.g., gates, obstacles, windows). The output of this module is a text string, $C_{mission}$, that encapsulates the high-level mission plan.

The core task of visual trajectory generation is achieved through a powerful generative model. This stage takes the standardized image I_{std} and the mission command $C_{mission}$ as conditional inputs. Our approach leverages the conditioning mechanism inherent in contemporary image-to-video (I2V) diffusion models, which generate video sequences from an initial frame and a text prompt. This capability is now standard in state-of-the-art systems such as Luma Ray2 [28], Google Veo 2 [29], Seedance 1.0 Pro [30], and Wan 2.2 [31]. For the purposes of this study, we selected the Wan 2.2 I2V Fast model, prioritizing its operational efficiency and high output fidelity. This model was chosen for its ability to generate high-quality, temporally consistent short videos V_{gen} conditioned on a starting image and a text prompt describing the desired motion (e.g., "a drone flying forward through a gate"). The average video generation time ranges from 20 to 60 s. This latency is currently acceptable for offline data synthesis and subtask-level planning, but it remains a bottleneck for fully real-time reactive navigation.

The generated video V_{gen} is processed using a monocular visual odometry module, ORB-SLAM3 [32], to reconstruct the camera motion and obtain a 3D trajectory estimate (Fig. 2). From this trajectory, state-action representations are derived based on the estimated pose evolution. These data are then used to train control policies for autonomous flight. In deployment, the drone executes the learned policy using real-time visual input, effectively closing the loop between synthetic data generation and real-world operation.

IV. Skill Repetition from Generated Videos

The reasoning block serves as the high-level planner in FlightDiffusion, mapping a user command and an input image to a structured flight plan. It employs a multi-modal LLM to perform scene understanding and task-conditioned planning. The process is as follows:

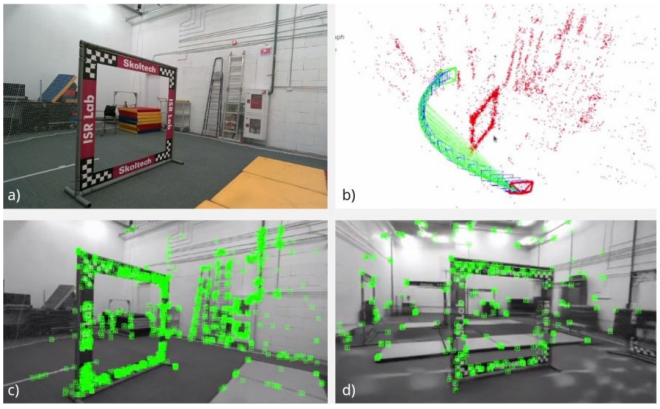

Fig. 2: Visual odometry pipeline for a synthetic video showing sequential processing for pose estimation and trajectory reconstruction: (a) initial frame, (b) ORB-SLAM tracking, (c) first-frame features, and (d) last-frame features.

TABLE I: UAV Hardware Configuration

Component	Specification
Onboard Computer	ASUS NUC, Intel Core i5-1240P, 16 GB RAM
Stereo Camera VIO	Intel RealSense T265, 848x800
Navigation Camera	Intel RealSense D455, 848x480
Flight Controller	SpeedyBee F405 V4
ESC	SpeedyBee BLS 55A
Motors	BrotherHobby Avenger 2810, 1180 KV

- **Input Processing**: The block receives a static image from the drone's onboard camera (or a simulated initial frame) and a natural language task description (e.g., "enter the restaurant through the door").
- **Contextual Analysis**: The image is encoded and provided to the LLM together with structured prompts, including a system prompt defining the planning role and output format, and a user prompt specifying the task.
- **Mission Generation**: The LLM extracts semantic information from the scene (e.g., obstacles, landmarks, and free space) and integrates it with the task objective to generate a step-by-step flight plan. The output includes waypoints, motion primitives, and maneuver descriptions, forming a structured representation of the desired trajectory.

Skill acquisition from generated videos is enabled by treating the pipeline output as a synthetic training source for the drone's control system. The process is as follows:

- **Video and Trajectory Generation**: The mission description produced by the reasoning block conditions a video diffusion model, which generates FPV video sequences representing candidate flight trajectories.
- **State-Action Extraction**: The generated video is processed with visual odometry to reconstruct a physically consistent 3D trajectory, from which state-action pairs are derived based on the estimated motion.
- **Policy Training**: These trajectories form a structured dataset of state-action pairs, enabling training of UAV

control policies via IL or RL.

- **Repetition and Generalization**: By sampling diverse initial conditions and task variations, the framework produces a wide range of trajectories, allowing the policy to learn robust and generalizable behaviors without extensive real-world data collection.

V. Trajectory and Action-Space Extraction for Mission Execution

In our method, the 3D trajectory is extracted from the synthetic monocular FPV video using ORB-SLAM3 operating in pure monocular mode. The system does not perform global mapping; instead, it reconstructs a local-consistent short-horizon camera motion. The main idea is to take a generated FPV video sequence, $V_{\text{gen}} = I_0, I_1, ..., I_N$, where I_t denotes the synthetic frame at time t, and convert it into a physically consistent 3D trajectory.

The process can be formalized as a function F mapping the video to a trajectory:

$$T_{\text{world}}^{\text{cam}}(t) = F(V_{\text{gen}}; \Theta_{\text{SLAM}}), \tag{1}$$

where $T_{\text{world}}^{\text{cam}}(t) \in SE(3)$ is the estimated camera pose (a rigid body transformation matrix) for frame I_t, and Θ_{SLAM} represents the parameters and state of the Simultaneous Localization and Mapping (SLAM) system (e.g., the map, keyframes, covisibility graph).

For each new synthetic frame I_t, ORB-SLAM3 performs the following key steps:

1) **Feature Extraction and Matching:** ORB features are extracted from I_t, resulting in a set of keypoints $K_t = \{\mathbf{k}_i\}$ with descriptors. These are matched against features in the current map M or a reference keyframe.
2) **Pose Optimization:** The camera pose is estimated by minimizing the reprojection error between matched 3D map points $\mathbf{X}_j \in \mathbb{R}^3$, and their corresponding 2D keypoints $\mathbf{k}_j$.

$$T^* = \arg\min_{T \in SE(3)} \sum_j \rho\left(\left\|\pi(T, \mathbf{X}_j) - \mathbf{k}_j\right\|_2^2\right), \tag{2}$$

where $\pi(\cdot)$ is the camera projection function and ρ is the robust kernel (e.g., Huber), to mitigate the effect of outliers.

3) **Map Point Triangulation and Bundle Adjustment:** New map points are triangulated from matched keyframe features, and local bundle adjustment optimizes keyframe poses and map point positions for consistent sparse reconstruction.

The output is a sequence of optimized camera poses $\{T_0, T_1, ..., T_N\}$ forming the trajectory $T_{\text{world}}^{\text{cam}}(t)$ as well as state-action pairs $(\boldsymbol{\xi}, \mathbf{v})$ for the drone to follow.

Algorithm 1 Trajectory Extraction from Synthetic Video

Input: Generated video V_{gen}
Output: Estimated trajectory $\{T_0, T_1, ..., T_N\}$, State-Actions$(\boldsymbol{\xi}, \mathbf{v})$
Initialize ORB-SLAM3 system (Monocular mode)
for each frame I_t in V_{gen} **do**
  $T_t \leftarrow$ ORB-SLAM3.Track(I_t) ▷ Solve Eq. 2
  Add T_t to trajectory
  StorePair$(\boldsymbol{\xi}_{\text{current}}, \mathbf{v}_t^c)$ ▷ Record the state-action pair
end for
return trajectory

The drone's state is continuously estimated in real-time using the OpenVINS[33] pipeline. The state of the drone at time t is defined by its pose in the world frame:

$$\boldsymbol{\xi}_t = [\mathbf{p}_t, \mathbf{q}_t]^T, \tag{3}$$

where $\mathbf{p}_t = [x, y, z]^T \in \mathbb{R}^3$ is the positional coordinate and $\mathbf{q}_t$ is the unit quaternion representing the orientation.

The trajectory $\{T_0, T_1, ..., T_N\}$ extracted from the synthetic video consists of a sequence of desired future poses, $T_i^{\text{des}} \in SE(3)$.

This trajectory is then passed on to the described controller to generate the velocity commands $\mathbf{v}^c$ that guide the physical drone. The complete navigation pipeline is given in Algorithm 2.

Algorithm 2 Full Navigation Pipeline

Input: Global Task Description, Initial Real Image I_0^{real}
Reason about task and generate subtask list L
for each subtask s_j **in** L **do**
  Reason s_j and image I^{real}
  Generate synthetic video V_{gen} for s_j
  Extract desired trajectory $\{T_i^{\text{des}}\}$ from V_{gen}
  for each T_i^{des} **in** trajectory **do**
    while drone has not reached T_i^{des} **do**
      Estimate current state $\boldsymbol{\xi}_t$ and velocity $\mathbf{v}^c$
      Send $\mathbf{v}^c$ to flight controller
    end while
  end for
  Take a new Image I_j^{real}
end for

VI. Experimental Evaluation

A. Trajectory Generation

To evaluate the quality of the generated video, we first extracted the trajectory from the synthetic sequence and then executed the corresponding flight using a real drone while recording video from its onboard camera. We then compared the generated and real visual streams and features. A qualitative comparison showed strong visual consistency between the generated video frames (Fig. 3), $I_{\text{gen}}[N]$, and the corresponding image captured by the camera onboard at the completion of the task, $I_{\text{real}}[N]$. The first and last frames exhibited similar semantic cues and geometric structures.

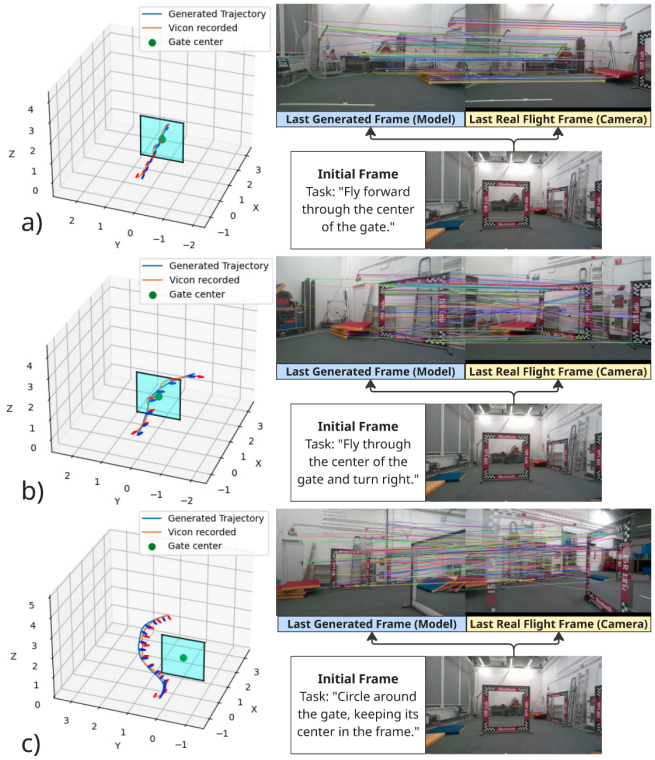

Fig. 3: Drone trajectory comparison: VICON recorded UAV trajectory (red) vs. synthetic video estimation trajectory (blue). Right: synthetic and real frames with ORB feature matches. Maneuvers: (a) center-gate flight, (b) right turn, (c) circumnavigation.

Since ORB-SLAM3 in monocular mode estimates motion up to an unknown similarity transform, the raw trajectory cannot be directly compared to the VICON data in metric units. For evaluation, we recover a global similarity transform:

$$S = (s, \mathbf{R}, \mathbf{t}) \in \mathrm{Sim}(3), \tag{4}$$

to align the generated trajectory with the VICON trajectory. The scalar s provides the metric scale, while $\mathbf{R}$ and $\mathbf{t}$ align orientation and origin. Translational errors in Table II are computed after this alignment.

For quantitative evaluation, we compared the aligned generated trajectory with the trajectory performed by the UAV, as measured by the VICON tracking system. Using a sequence of K corresponding poses, the translational error at each step k was defined as the Euclidean distance between the estimated position $\mathbf{p}_{\mathrm{gen}}[k] = (x_k, y_k, z_k)$, while the rotational component was represented by its Tait-Bryan angles (yaw, pitch, roll) $\boldsymbol{\theta}_{\mathrm{gen}}[k] = (\psi_k, \theta_k, \phi_k)$ and the ground-truth position $\mathbf{p}_{\mathrm{gt}}[k] = (x_{gt,k}, y_{gt,k}, z_{gt,k})$ and orientation $\boldsymbol{\theta}_{\mathrm{gt}}[k] = (\psi_{gt,k}, \theta_{gt,k}, \phi_{gt,k})$:

$$e_{\mathrm{trans}}[k] = \left\| \mathbf{p}_{\mathrm{gen}}[k] - \mathbf{p}_{\mathrm{gt}}[k] \right\|_2. \tag{5}$$

The orientation error was quantified as:

$$e_{\mathrm{rot}}[k] = \left\| \boldsymbol{\theta}_{\mathrm{gen}}[k] - \boldsymbol{\theta}_{\mathrm{gt}}[k] \right\|_2. \tag{6}$$

TABLE II: Quantitative results of trajectory tracking.

Metric	**Translation** (m)	**Rotation** (rad)
Mean Error	0.25	0.19
Max Error	0.44	0.51
RMSE	0.28	0.24

The aggregated results from three experimental trials are presented in Table II. The low values for all error metrics demonstrate that the generated trajectory closely matches the actual flight path, with an average positional RMSE of 0.28 m and an orientation RMSE of 0.24 rad.

B. Scale Ambiguity and Trajectory Error Analysis

Let s^* denote the true global scale and $\hat{s}$ the estimated scale. If the monocular trajectory is $\hat{\mathbf{p}}_k$ in arbitrary units, the metric estimate is:

$$\mathbf{p}_k^{\mathrm{est}} = \hat{s}\,\hat{\mathbf{p}}_k, \tag{7}$$

whereas the true metric position is:

$$\mathbf{p}_k^{\mathrm{gt}} = s^*\,\hat{\mathbf{p}}_k, \tag{8}$$

under an ideal shape reconstruction. The resulting translation error induced purely by scale error is:

$$\mathbf{e}_k^{\mathrm{scale}} = (\hat{s} - s^*)\,\hat{\mathbf{p}}_k. \tag{9}$$

Taking the Euclidean norm gives:

$$\|\mathbf{e}_k^{\mathrm{scale}}\|_2 \approx \frac{|\hat{s} - s^*|}{s^*}\,\|\mathbf{p}_k^{\mathrm{gt}}\|_2. \tag{10}$$

This relationship shows that the translational error grows approximately linearly with the distance traveled from the alignment origin. Consequently, part of the translation RMSE reported in Table II and the drift trend observed in Fig. 3 may arise from uncertainty in the global scale estimate rather than solely from local pose reconstruction errors.

The remaining errors can be primarily attributed to the stochastic nature of the diffusion process in video generation, which introduces variability in the synthesized trajectories. In particular, the model may produce hallucinated elements that are not present in the initial scene, especially when the prompt references objects or structures that are weakly supported by the visual input. We observed that such artifacts can slightly distort the predicted trajectory, typically leading to unnecessary avoidance maneuvers or brief tracking of non-existent targets.

To mitigate this effect, the visual odometry pipeline can be augmented with a safety filtering stage based on point-cloud and visual analysis derived from visual odometry feature observations of the initial frame. By comparing the reconstructed scene structure with the generated content, features associated with hallucinated elements can be identified and removed. This filtering step helps ensure that the resulting trajectory remains consistent with the actual environment and reduces the impact of diffusion-induced artifacts on downstream control.

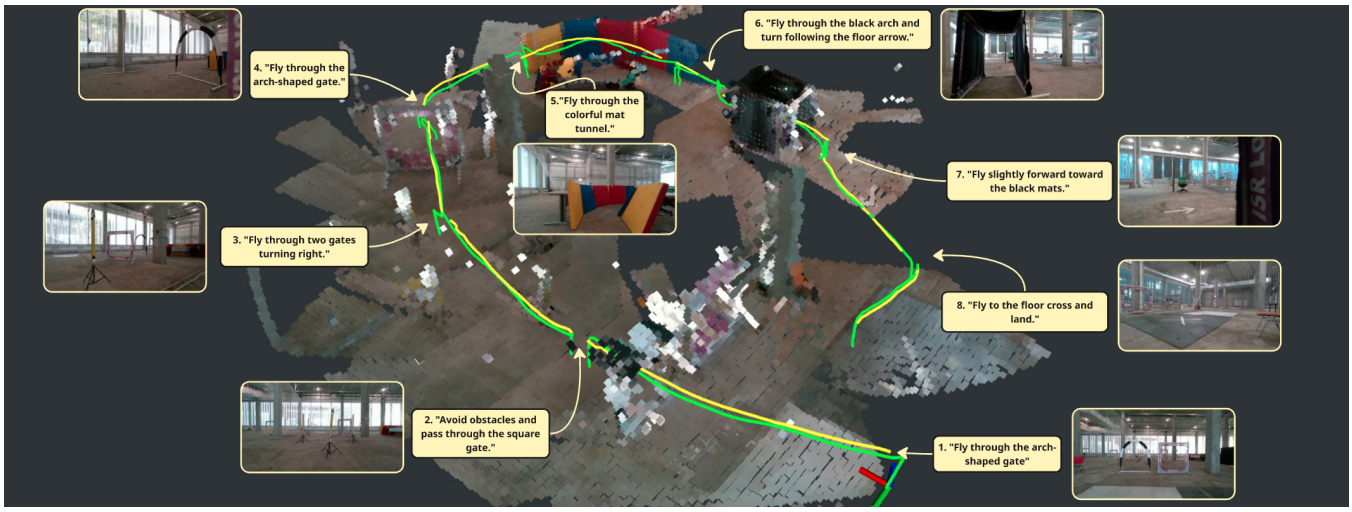

Fig. 4: Long-horizon real-world experiment. The point-cloud map of the flight arena is shown with the executed UAV trajectory (green) and task-conditioned trajectories for successive subtasks (yellow). After each subgoal, the UAV captures an image and generates the corresponding reasoning task for video generation to complete the global task.

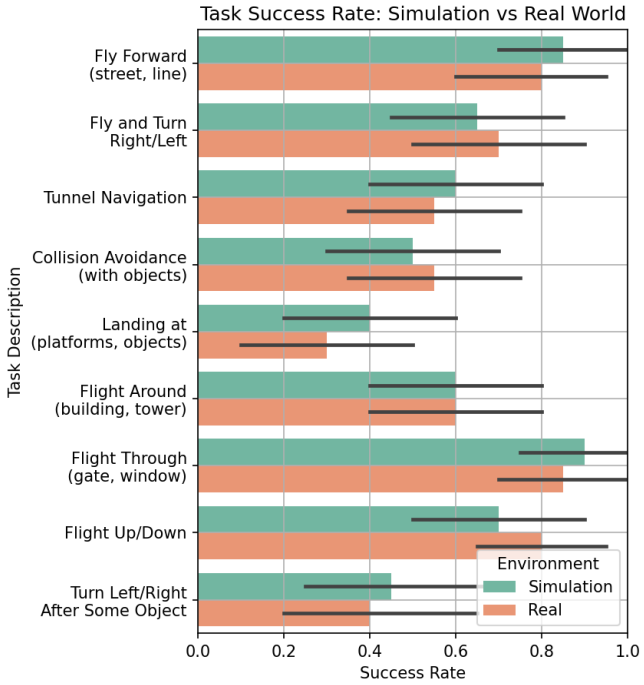

Fig. 5: Success rates for nine maneuver types in simulation and real-world experiments.

C. Simulation vs. Real-World Maneuver Evaluation

To evaluate the transfer of autonomous flight performance from simulation to reality, each of nine distinct maneuvers (including forward flight, tunnel navigation, collision avoidance, and landing) were executed twenty times in both a simulated and a real-world environment, yielding a success rate for each condition (Fig. 5). For the simulation experiments, we used the Small City world [34] and an office environment [35] in the Gazebo simulation. A two-way analysis of variance (ANOVA) without replication was

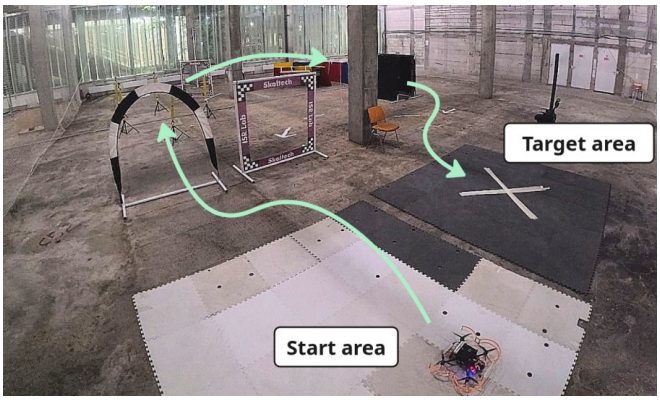

Fig. 6: The experimental UAV platform in a flight arena.

conducted to assess the effects of the environment (simulation vs. reality) and the maneuver type on the success rates. The analysis revealed no statistically significant main effect of the environment ($F(1,16) = 0.394, p = 0.541$), indicating that the overall mean success rate in simulation ($M = 0.628, SD = 0.162$) was not significantly different from that in reality ($M = 0.617, SD = 0.177$). However, a highly significant main effect of maneuver type was found ($F(8,16) = 26.250, p < 0.001$), confirming that the intrinsic difficulty varied substantially across the different flight tasks. A paired-sample t-test corroborated the ANOVA finding, showing no significant difference between the paired conditions ($t(8) = 0.318, p = 0.758$). These results demonstrate that while the complexity of maneuvers significantly influences performance, the simulation serves as a highly valid environment, with overall system performance being statistically equivalent to real-world operation.

D. Long Horizon Task

Objective. The primary objective of the experiment was to evaluate the drone's ability to autonomously complete a complex long-horizon navigation course in simulated (Fig. 7, 8)

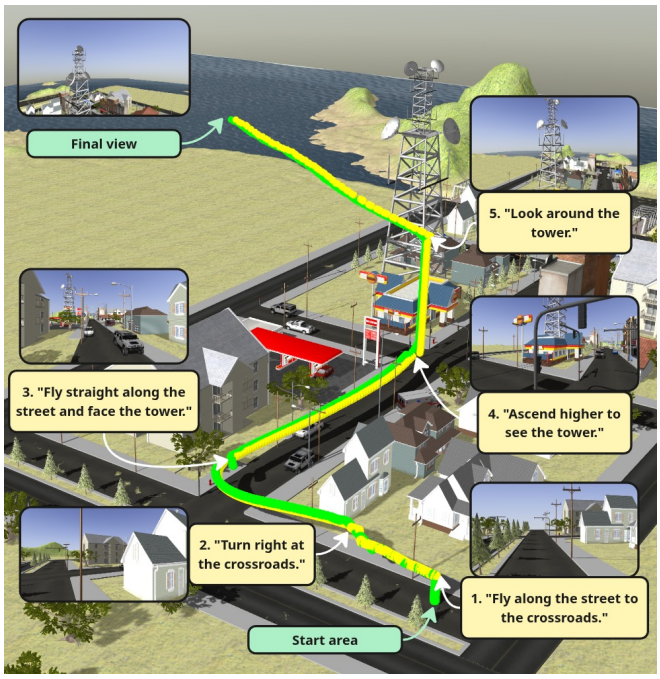

Fig. 7: Simulation environment for long-horizon tower inspection. Green: ArduPilot SITL trajectory; yellow: generated trajectories. Input images and task specifications are shown for each planning stage.

and physical real-world (Fig. 5, 6) testing environments.

Global Task Definition. The global long-horizon task prompt was defined as follows:

- Simulated outdoor environment (Fig. 7). "The drone navigates over the city, proceeds to the first crossroads, makes a right turn, locates the tower, ascends, and performs an inspection of the structure".
- Simulated indoor environment (Fig. 8). "The task involves exploring an office environment to locate a wooden door and then navigating to exit the room through it".
- Real-world environment (Fig. 6). "Navigate through two gates, pass an arch, avoid yellow obstacles, make a right turn, cross a square and an arch gate, fly between colored mats, traverse a black tunnel, follow a ground arrow and land on a white cross".

Setup. The global task was decomposed into steps such as: Navigate to Arch Gate 1, Avoid Yellow Obstacles, Turn Right, etc. For each subtask, the model performed reasoning based on the current FPV camera image (see Fig. 4). A video trajectory was generated for each step, and keyframes were processed to extract the trajectory. After completing a subtask (e.g., passing the first arch gate), the drone state was updated and a new image captured. The drone then switched to hover mode while waiting for the next trajectory.

Results. FlightDiffusion successfully completed the long-horizon mission in 5 min 20 s in the real-world environment, 4 min in the outdoor simulation, and 2 min 10 s in the indoor office simulation, with an average speed of 1 m/s in the real-world trial. Each subtask required about 30 s to generate the synthetic video. The generated trajectories were realistic and physically feasible, and could be reliably converted into executable flight paths. Figs. 4, 7, and 8 show the generated trajectories (yellow) and UAV trajectories (green).

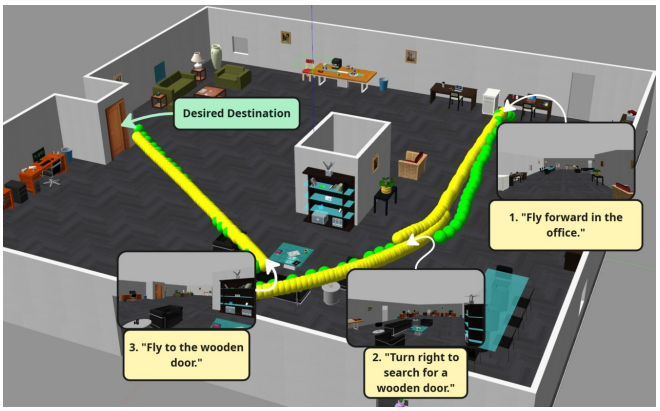

Fig. 8: Simulated office environment designed for long-horizon tasks involving locating and reaching a door.

VII. Conclusion and Future Work

This paper introduced FlightDiffusion, a diffusion-model-based framework for autonomous drone navigation and dataset generation that integrates generative video synthesis with trajectory reconstruction. The proposed approach addresses key challenges in data-driven aerial robotics, particularly the limited availability of large-scale training data and the sim-to-real transfer gap.

The core strength of FlightDiffusion lies in its dual capability. It serves both as a learning pipeline tool, enabling reasoning-driven navigation from a single FPV frame, and as a scalable data generation pipeline. The framework synthesizes diverse FPV trajectories and reconstructs corresponding state-action representations, providing a cost-effective approach to dataset creation. This is supported by quantitative results showing low reconstruction error, with a positional RMSE of 0.28 m and an orientation RMSE of 0.24 rad, indicating that the generated trajectories are physically consistent and executable. Evaluation across simulated and real-world environments shows no statistically significant difference in performance (ANOVA: $(F(1,16) = 0.394, p = 0.541)$; paired t-test: $(t(8) = 0.318, p = 0.758)$), suggesting that policies trained on synthetic data can transfer effectively under the evaluated conditions. Additionally, the effect of maneuver type $(F(8,16) = 26.250, p < 0.001)$ indicates that task difficulty varies across scenarios and is reflected in the system's performance.

A limitation of the current approach arises from monocular reconstruction, which introduces scale ambiguity and may lead to drift over longer trajectories. In addition, the stochastic nature of diffusion-based video generation can produce hallucinated or weakly grounded elements, leading to minor deviations in the reconstructed trajectory. While the resulting trajectories remain physically consistent and executable, fu-

ture work will focus on incorporating additional constraints, such as depth estimation or scene-consistent filtering, to improve robustness. In summary, FlightDiffusion provides a practical approach for linking high-level task reasoning with low-level control through diffusion-based video generation and trajectory reconstruction. The framework enables scalable dataset generation and supports the development of robust navigation policies. Future work will focus on reducing inference latency, expanding environmental diversity, and incorporating additional sensory modalities to further improve generalization and real-time applicability.

ACKNOWLEDGMENTS

Research reported in this publication was financially supported by the RSF grant No. 24-41-02039.

REFERENCES

[1] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Mueller, V. Koltun, and D. Scaramuzza, "Champion-level drone racing using deep reinforcement learning," *Nature*, vol. 620, pp. 982–987, Sept. 2023.

[2] T. Elmokadem and A. V. Savkin, "A Hybrid Approach for Autonomous Collision-Free UAV Navigation in 3D Partially Unknown Dynamic Environments," *Drones*, vol. 5, no. 3, July 2021.

[3] V. J. Hodge, R. Hawkins, and R. Alexander, "Deep reinforcement learning for drone navigation using sensor data," *Neural Computing and Applications*, vol. 33, no. 6, pp. 2015–2033, June 2021.

[4] X. Wang, Z. Ning, S. Guo, M. Wen, L. Guo, and H. V. Poor, "Dynamic UAV deployment for differentiated services: A multi-agent imitation learning based approach," *IEEE Transactions on Mobile Computing*, vol. 22, no. 4, pp. 2131–2146, Sept, 2021.

[5] A. Das, R. D. Yadav, S. Sun, M. Sun, S. Kaski, and W. Pan, "DroneDiffusion: Robust Quadrotor Dynamics Learning with Diffusion Models," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 19-23, 2025, pp. 1604–1610.

[6] Y. Zhong, A. Zhao, T. Wu, T. Zhang, and F. Gao, "Automatic Generation of Aerobatic Flight in Complex Environments via Diffusion Models," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Oct. 19-25, 2025, pp. 21 235–21 242.

[7] O. Sautenkov, A. Akhmetkazy, Y. Yaqoot, M. A. Mustafa, G. Tadevosyan, A. Lykov, V. Serpiva, and D. Tsetserukou, "UAV-VLPA*: Vision-Language Guided Global-Local UAV Mission Planning from Satellite Imagery," in *Proc. IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, Dec. 3-7, 2025, pp. 2354–2359.

[8] V. Serpiva, A. Lykov, A. Myshlyaev, M. H. Khan, A. A. Abdulkarim, O. Sautenkov, and D. Tsetserukou, "RaceVLA: VLA-based Racing Drone Navigation with Human-like Behaviour," 2025, arXiv:2503.02572.

[9] J. Xiao, C. W. Tsao, Y. Zhang, and M. Feroskhan, "FM-Planner: Foundation Model Guided Path Planning for Autonomous Drone Navigation," 2025, arXiv:2505.20783.

[10] "Context-Aware Autonomous Drone Navigation Using Large Language Models (LLMs)," in *Proc. of the AAAI Symposium Series*, Aug. 2025, pp. 102–107.

[11] H. Chen, Y. Tang, A. Tsourdos, and W. Guo, "Contextualized Autonomous Drone Navigation Using LLMs Deployed in Edge-Cloud Computing," in *Proc. Int. Conf. on Machine Learning and Autonomous Systems (ICMLAS)*, Mar. 10-12, 2025, pp. 1373–1378.

[12] V. Serpiva, A. Lykov, J. Sam, A. Fedoseev, and D. Tsetserukou, "Diffusioncinema: Text-to-aerial cinematography," in *Proc. ACM/IEEE Int. Conf. on Human-Robot Interaction*, 2026, pp. 52—56.

[13] O. Sautenkov, Y. Yaqoot, A. Lykov, M. A. Mustafa, G. Tadevosyan, A. Akhmetkazy, M. A. Cabrera, M. Martynov, S. Karaf, and D. Tsetserukou, "UAV-VLA: Vision-Language-Action System for Large Scale Aerial Mission Generation," in *Proc. ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI)*, Mar. 04-06, 2025, pp. 1588–1592.

[14] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters, "An algorithmic perspective on imitation learning," *Foundations and Trends in Robotics*, vol. 7, no. 1–2, pp. 1–179, 2018.

[15] M. S. Tayar, L. K. de Oliveira, F. A. G. Tommaselli, J. D. Negri, T. H. Segreto, R. V. Godoy, and M. Becker, "Autonomous UAV Flight Navigation in Confined Spaces: A Reinforcement Learning Approach," in *Latin American Robotics Symposium (LARS)*, Oct. 2025, pp. 1–6.

[16] P. Wei, P. Ragbir, S. G. Vougioukas, and Z. Kong, "Vision-based navigation of unmanned aerial vehicles in orchards: An imitation learning approach," *Computers and Electronics in Agriculture*, vol. 238, p. 110802, Nov. 2025.

[17] K. Kondo, A. Tagliabue, X. Cai, C. Tewari, O. Garcia, M. Espitia-Alvarez, and J. P. How, "CGD: Constraint-Guided Diffusion Policies for UAV Trajectory Planning," 2024, arXiv:2405.01758.

[18] I. Zhura, S. Karaf, F. Batool, N. D. W. Mudalige, V. Serpiva, A. A. Abdulkarim *et al.*, "SwarmDiffusion: End-To-End Traversability-Guided Diffusion for Embodiment-Agnostic Navigation of Heterogeneous Robots," 2025, arXiv:2512.02851.

[19] M. Seo, Y. Cho, Y. Sung, P. Stone, Y. Zhu, and B. Kim, "PRESTO: Fast Motion Planning Using Diffusion Models Based on Key-Configuration Environment Representation," in *Proc. Int. Conf. on Robotics and Automation (ICRA)*, May 19-23, 2025, pp. 10 861–10 867.

[20] S. Yang, L. Liu, B. Chen, S. Cheng, Z. Shi, and Z. Zou, "Goodflight: Goal-oriented diffusion model for flight trajectory prediction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 61, no. 3, pp. 7447–7465, 2025.

[21] D. Xing and A. Tzes, "Synthetic Aerial Dataset for UAV Detection via Text-to-Image Diffusion Models," in *Proc. Conf. on Artificial Intelligence (CAI)*, 2023, pp. 51–52.

[22] M. Chen, H. Chu, and X. Wei, "Flocking Control Algorithms Based on the Diffusion Model for Unmanned Aerial Vehicle Systems," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 3, pp. 1271–1282, April 2021.

[23] S. Liu, H. Zhang, Y. Qi, P. Wang, Y. Zhang, and Q. Wu, "AerialVLN: Vision-and-Language Navigation for UAVs," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Oct. 1-6, 2023, pp. 15 384–15 394.

[24] J. Zhang, A. Li, Y. Qi, M. Li, J. Liu, S. Wang *et al.*, "Embodied navigation foundation model," 2025, arXiv:2509.12129.

[25] J. Ye, S. Papaioannou, and P. Kolios, "VLM-RRT: Vision language model guided RRT search for autonomous uav navigation," in *Proc. IEEE Int. Conf. on Unmanned Aircraft SystemS (ICUAS)*, May 14-17, 2025, pp. 633–640.

[26] L. Suomela, S. K. Arachchige, G. F. Torres, H. Edelman, and J.-K. Kämäräinen, "Synthetic vs. Real Training Data for Visual Navigation," 2025, arXiv:2509.11791.

[27] Google DeepMind, "Gemini 2.5 flash," https://deepmind.google/technologies/gemini/, 2024.

[28] Luma Labs, "Ray2: Image-to-video model," 2025, accessed: Sep. 16, 2025. [Online]. Available: https://lumalabs.ai/ray

[29] Google DeepMind, "Veo 2: Image-to-video," 2024, accessed: Sep. 16, 2025. [Online]. Available: https://blog.google/technology/google-labs/video-image-generation-update-december-2024/

[30] ByteDance, "Seedance 1.0 pro: Image-to-video," 2024, accessed: Sep. 16, 2025. [Online]. Available: https://seed.bytedance.com/en/seedance

[31] Replicate, "Wan 2.2 image-to-video fast," 2025, accessed: Sep. 16, 2025. [Online]. Available: https://replicate.com/wan-video/wan-2.2-i2v-fast

[32] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, May 25, 2021.

[33] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A Research Platform for Visual-Inertial Estimation," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 31, 2020, pp. 4666–4672.

[34] ARTI-Robots, "gazebo-worlds: Outdoor environments for robotic simulation," https://github.com/ARTIRobots/gazebo_worlds, 2019.

[35] A. Rasouli and J. K. Tsotsos, "The Effect of Color Space Selection on Detectability and Discriminability of Colored Objects," 2017, arXiv:1702.05421.