Automatic Generation of Aerobatic Flight in Complex Environments via Diffusion Models

Yuhang Zhong[1,2], Anke Zhao[1,2], Tianyue Wu[1,2], Tingrui Zhang[1,2] and Fei Gao[1,2,*]

Fig. 1: Our method enables automatic generation of successive long-horizon aerobatic maneuvers, allowing drones to traverse through a complex industrial factory with dynamically feasible motion.

Abstract—**Performing striking aerobatic flight in complex environments demands manual designs of key maneuvers in advance, which is intricate and time-consuming as the horizon of the trajectory performed becomes long. This paper presents a novel framework that leverages diffusion models to automate and scale up aerobatic trajectory generation. Our key innovation is the decomposition of complex maneuvers into aerobatic primitives, which are short frame sequences that act as building blocks, featuring critical aerobatic behaviors for tractable trajectory synthesis. The model learns aerobatic primitives using historical trajectory observations as dynamic priors to ensure motion continuity, with additional conditional inputs (target waypoints and optional action constraints) integrated to enable user-editable trajectory generation. During model inference, classifier guidance is incorporated with batch sampling to achieve obstacle avoidance. Additionally, the generated outcomes are refined through post-processing with spatial-temporal trajectory optimization to ensure dynamical feasibility. Extensive simulations and real-world experiments have validated the key component designs of our method, demonstrating its feasibility for deploying on real drones to achieve long-horizon aerobatic flight.**

I. INTRODUCTION

Aerobatic freestyle flight in complex environments stands as one of the most striking and visually impressive drone-based extreme sports [1]. By planning safe but agile maneuvers and incorporating creative combinations of these highly dynamic movements, one can enable spectacular flight effects that captivate spectators alike. However, the design process can be intricate, as multiple competing requirements including obstacle avoidance, dynamic feasibility and visual impact must be simultaneously satisfied. Previous works address this problem by manually adjusting trajectory parameters such as waypoints [2]–[5] with trajectory optimization. Unfortunately, these methods remain constrained by their heavy reliance on laborious parameter tuning and domain-specific expertise in aerobatic flight, creating substantial barriers for non-expert operators attempting to design even basic aerobatic maneuvers. To bridge this gap, this paper introduces an efficient framework for the automatic generation of diverse aerobatic flights, enabling practitioners to design complex long-horizon multi-maneuver trajectories with minimal human intervention.

*Corresponding Author: Fei Gao.

[1]Institute of Cyber-Systems and Control, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China.

[2]Huzhou Institute of Zhejiang University, Huzhou 313000, China.

E-mail:{YuhangZhong, AnkeZhao, tianyueh8erobot, tingruizhang, fgaoaa}@zju.edu.cn

Recently, diffusion models have shown remarkable capacity to capture multi-modal distributions, enabling diverse generation in human motion synthesis [6]–[8], and trajectory planning [9]–[11], where models produce realistic motion sequences with diverse styles and specialized behaviors. Building on these capabilities, we explore their potential for aerobatic generation. An intuitive idea is to learn directly from long-horizon aerobatic demonstrations. However, such high-quality data remains scarce due to labor-intensive manual design and time-consuming generation processes. While some works [12]–[15] attempt to realize long-horizon generation by combining short-horizon sequences, they face twofold critical challenges when applied in aerobatic scenarios. First, aerobatic maneuvers demand strict sequential pose transitions (e.g., continuous 360° z-axis rotation in a loop maneuver). When trajectories are fragmented into short segments for training, the model fails to capture the precise timing and coordination between sequential poses, resulting in incomplete motion generation. Second, unlike image generation [16] where pixel-level discontinuities are visually tolerable, aerobatic flight demands strict spatial-temporal continuity. Naive concatenation of short motion segments inevitably introduces visible discontinuities, which is a critical flaw given the highly dynamic nature of maneuvers requiring seamless state evolution. Consequently, it's crucial to define a modular yet kinematically consistent representation for achieving seamless composition of long-horizon aerobatic maneuvers.

In this paper, we propose to learn from aerobatic primitives to address the aforementioned challenges. An aerobatic primitive is a sequence of maneuver frames that captures key attitude changes over time and can be seamlessly combined with other primitives to achieve successive and arbitrarily long-horizon aerobatic flight. Crucially, these primitives support explicit conditioning on both maneuver styles and target waypoints, enabling user-specific trajectory generation through intuitive parameter adjustment. However, without the awareness of the previous executed primitives, the continuity of the aerobatic primitives is hard to guarantee. To mitigate this, we incorporate historical trajectory observations as transitional priors into the model architecture, allowing it to capture the latent dynamics underlying primitive transitions. While generating high-quality motions, the model is not trained with environmental information, thus providing no collision avoidance guarantees in unseen environments. We address this problem by adopting batch sampling for each primitive generation with classifier guidance [17], [18], a widely used technique for steering the generation toward a specific target distribution. The coarse collision check on the generated trajectories is applied in each inference to further improve the obstacle avoidance success rate.

While diffusion models can generate robot-executable trajectories via positional or velocity control [19], [20], they fail to meet the demands of precise control over actuator-level commands (e.g., thrust and angular velocities) during aerobatic flight. Although models implicitly encode such control signals during training, they lack explicit enforcement of dynamic feasibility, which is critical for successful flight in practical deployment. Therefore, post-processing with trajectory optimization is proposed to ensure the final trajectory stays within dynamic constraints. Notably, due to the extreme nonlinearity associated with optimizing attitude and angular velocity in the differential flatness based trajectory optimization framework [21]–[23], we design a hierarchical optimization framework to guide the final optimization to converge to a favorable local optimum, making practical deployment feasible.

Our contributions are summarized as follows:

1) By learning from aerobatic primitives and incorporating an additional collision avoidance strategy, our diffusion model is capable of generating arbitrary long-horizon trajectories in complex environments despite being trained exclusively on short-horizon demonstration.

2) Post-processing with hierarchical trajectory optimization is designed to guarantee that generated aerobatic trajectories are physically feasible.

3) The simulation and experimental results demonstrate that the proposed method exhibits a high capability of generating a wide variety of aerobatic trajectories in complex environments.

II. RELATED WORKS

A. Aerobatic Flight Generation for Quadrotors

Generating aerobatic flight presents significant challenges due to its competing requirements for rapid attitude changes and dynamic feasibility planning. Current approaches can be broadly categorized into two paradigms. Rule-based approaches [2], [4], [24], [25] employ motion decomposition strategies, in which complex maneuvers are segmented into different phases. While Kaufmann et al. [2] and Lu et al. [4] utilize vertical circles or arcs to enable basic 3D aerobatic motion generation beyond planar constraints, they suffer from limited adaptability to dynamic environments and require laborious parameter tuning for each specific maneuver. As the optimization method demonstrates significant success in quadrotor applications [26]–[29], growing research formulates aerobatic generation as trajectory optimization problems to leverage their inherent flexibility. The authors in [3], [5] achieve aerobatic trajectory generation of tail-sitter by adjusting positional- and temporal-related parameters, enabling multi-maneuver flights in open indoor and outdoor environments. However, existing methods primarily focus on aerobatic trajectory generation in open environments, neglecting essential obstacle interactions. More critically, these approaches require meticulously designed initial values to circumvent suboptimal local minima, a critical limitation stemming from the nonconvex optimization landscape created by strong nonlinearities in coupled attitude-obstacle constraints.

B. Diffusion Model for Motion Generation

Diffusion models have emerged as a widely adopted approach for generating motions across diverse applications.

In motion planning, researchers utilize diffusion models to produce trajectories characterized by optimal distributions of positions and velocities. To enforce task-specific constraints, reinforcement learning (RL) rewards [9] or task-oriented cost functions [10], [11] are integrated to guide trajectory distribution refinement. For human motion synthesis, the diffusion model's capacity for modeling high-dimensional spaces enables learning intricate motion representations. Additional conditional inputs, such as text-guided motion styles [6] and partial state constraints [30] for motion generation, further enhance the editing flexibility of synthesized motions. However, the above methods primarily focus on generating fixed-length motion sequences, leaving the potential of diffusion models for long-horizon tasks underexplored. While recent studies employ policy-based methods [19], [20] to generate action sequences in manipulation and visual navigation tasks, their reliance on the Markov assumption often leads to myopic generation behaviors. This manifests as delayed responses to impending obstacles and fragmented execution of aerobatic maneuvers. In contrast to these approaches, our work proposes a novel framework that learns aerobatic primitives that capture key aerobatic maneuver dynamics. By strategically combining these primitives with guidance design, we achieve coherent long-horizon aerobatic motion generation while preserving consistency with physical constraints and environmental interactions.

III. AEROBATIC DIFFUSION MODEL

A. Aerobatic Primitive Representation

Aerobatic primitives are expressed as a sequence of states $\boldsymbol{\tau} = \{\boldsymbol{x}_0, \boldsymbol{x}_1 \cdots, \boldsymbol{x}_{N_a}\}$, $\boldsymbol{x}_i = \{s, \boldsymbol{p}, \boldsymbol{r}\} \in \mathbb{R}^{10}$ with a constant time step, where $\boldsymbol{p} \in \mathbb{R}^3$ is the position of the quadrotor and $\boldsymbol{r} \in \mathbb{R}^6$ denotes a continuous 6-DoF rotation representation [31]. Notably, different maneuvers possess distinct execution durations, resulting in discrete state sequences with non-uniform lengths. This conflicts with the inherent fixed-length requirement of the diffusion model's output. To address this, a state flag $s \in \{0, 1\}$ is introduced to dynamically truncate the results when s transitions from 0 to 1, where $s = 0$ indicates the confidence that the current state belongs to the actual maneuver, while $s = 1$ represents padding states. This simple design allows variable-length primitive generation while maintaining fixed network output dimensions. To ensure complete motion generation, we set the output sequence length N_a to accommodate the maximum primitive duration in our dataset, with shorter sequences naturally terminated through s-guided truncation.

B. Data Preparation

We generate an expert dataset through an optimization-based method, focusing on short-horizon aerobatic maneuvers in open space. As Figure 2 demonstrates, given that the generated trajectory is modeled as a continuous polynomial, aerobatic primitives are obtained by sampling from the specific segments of the complete trajectory. To facilitate dynamic transitions, we strategically prepend or append redundant trajectory segments to each primitive. This

Fig. 2: Illustration of aerobatic primitive generation, the trajectory segments containing the aerobatic maneuver are segmented, and the discretized motion sequences are sampled from it. Redundant trajectory segments are added to simulate the transition between aerobatic primitives.

also benefits the model conditioning, as motion sequences from previous trajectory segments can serve as prior observations for learning seamless transitions. Additionally, the end state of aerobatic primitives is treated as a target waypoint and is incorporated into the model as a condition to enhance controllability. Different aerobatic maneuvers are randomly sampled based on predefined maneuver design rules. Notably, the generated demonstrations are inherently environment-agnostic by design. During model inference, we dynamically incorporate environmental context to enable obstacle-aware trajectory generation, as detailed in Section III-E.

C. Conditional Diffusion Model for Aerobatic Primitive Generation

We propose the Aerobatic Diffusion Model (AeroDM), a conditional diffusion model [17], [18], to generate aerobatic primitives. This model generates the samples by learning the denoising process $p_\theta(\boldsymbol{\tau}^{t-1}|\boldsymbol{\tau}^t, \boldsymbol{c})$ from pure Gaussian noise $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ to the original data distribution $p(\boldsymbol{\tau}^0|\boldsymbol{c})$ under the special conditions $\boldsymbol{c}$. In this paper, $\boldsymbol{c}$ contains previous state observations, target waypoint $\boldsymbol{p}_t$ which indicates the terminal position of $\boldsymbol{\tau}$, and action a that presents the aerobatic maneuver style. The denoising process is the reverse of the forward process $q(\boldsymbol{\tau}^t|\boldsymbol{\tau}^{t-1}, \boldsymbol{c})$, which corrupts the data structure by gradually adding increasing noise. The predicted sample distribution can be expressed as:

$$p_\theta(\boldsymbol{\tau}^0|\boldsymbol{c}) = \int p(\boldsymbol{\tau}^T|\boldsymbol{c}) \prod_{t=1}^{T} p_\theta(\boldsymbol{\tau}^{t-1}|\boldsymbol{\tau}^t, \boldsymbol{c}) d\boldsymbol{\tau}^{1:T}, \quad (1)$$

where $p(\boldsymbol{\tau}^T) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. During training, the gaussian noise is added in the forward process:

$$q(\boldsymbol{\tau}^t|\boldsymbol{\tau}^{t-1}, \boldsymbol{c}) = \mathcal{N}(\sqrt{\alpha_t}\boldsymbol{\tau}^{t-1}, (1 - \alpha_t)I), \quad (2)$$

where $\alpha_t \in (0, 1)$ are predefined scheduled parameters. Instead of predicting diffusion noise ϵ, we choose to directly predict the original sample $\boldsymbol{\tau}^0$ to facilitate explicit geometric constraint integration. The reconstruction loss can be written as:

$$\mathcal{L}_{recon} = E_{\boldsymbol{\tau}_{data} \sim p(\boldsymbol{\tau}_{data}), t \sim [1,T]}[\|\boldsymbol{\tau}_{data} - \boldsymbol{\tau}_\theta(\boldsymbol{\tau}^t, t)\|_2^2].$$
$$(3)$$

Fig. 3: The architecture of the diffusion process. (A) Schematic of the overall process. (B) Detailed structure of the Aerobatic Diffusion Model.

Inspired by work [30], we introduce velocity loss to encourage smooth transition along the primitives:

$$\mathcal{L}_{vel} = \frac{1}{N_a - 1} \sum_{i=1}^{N_a - 1} \left\| (\boldsymbol{x}_i^\theta - \boldsymbol{x}_{i-1}^\theta) - (\boldsymbol{x}_i^{data} - \boldsymbol{x}_{i-1}^{data}) \right\|_2^2.$$

(4)

The aerobatic generation process, illustrated in Fig. 3(a), operates iteratively through the aerobatic diffusion model. At each process step i, the model generates the current aerobatic primitive τ conditioned on target waypoints, and optional action signals with environment-related guidance. This process continues until the trajectory sequence $\{..., \tau_{i-1}, \tau_i\}$ reaches predefined aerobatic maneuver number N_{aero}. After generation, the primitives are concatenated and refined in the post-processing stage. Notably, the action input can be omitted during inference to enable diverse style generation.

D. Network Architecture

As illustrated in Fig. 3(B), we adopt a Diffusion Transformer architecture to model the temporal dependencies in aerobatic primitive sequences. The model utilizes a decoder-only transformer backbone where both the denoising trajectory τ^t (at diffusion time step t) and historical observations of previous primitives τ^{t-1} are jointly processed through self-attention layers. This design explicitly enforces continuity between the generated primitive τ^t and its predecessors. Additionally, conditional inputs including the denoising timestamp t, target waypoint $\boldsymbol{p}_t$, and action a, are first encoded via MLP embeddings (φ) separately and then integrated into the transformer through a cross-attention module.

E. Collision Avoidance Strategy

The generated aerobatic primitives have no guarantee of collision avoidance in the cluttered environment. We mitigate this problem by adding cost guidance for obstacle avoidance when doing model inference. This technique is derived from the conditional probability with Bayes Rule:

$$p(\boldsymbol{\tau}^{t-1}|\boldsymbol{\tau}^t, O) \propto p(\boldsymbol{\tau}^{t-1}|\boldsymbol{\tau}^t)p(O|\boldsymbol{\tau}^{t-1}), \quad (5)$$

where $p(\boldsymbol{\tau}^{t-1}|\boldsymbol{\tau}^t)$ is the denoising process, and $p(O|\boldsymbol{\tau}^{t-1})$ is the likelihood of achieving collision avoidance. By following the derivation in the work of [10], the result can be approximated as Gaussian:

$$p(\boldsymbol{\tau}^{t-1}|\boldsymbol{\tau}^t, O) \approx \mathcal{N}(\boldsymbol{\tau}^{t-1}, \mu + \Sigma g, \Sigma), \quad (6)$$

where μ and Σ are mean and variance of $p(\boldsymbol{\tau}^{t-1}|\boldsymbol{\tau}^t)$, g denotes an energy function:

$$\boldsymbol{g} = \nabla_{\boldsymbol{\tau}^{t-1}} \log p(O|\boldsymbol{\tau}^{t-1})|_{\boldsymbol{\tau}^{t-1}=\boldsymbol{\mu}} \quad (7)$$
$$= \sum_i \lambda_i \nabla_{\boldsymbol{\tau}^{t-1}} c_i(\boldsymbol{\tau}^{t-1})|_{\boldsymbol{\tau}^{t-1}=\boldsymbol{\mu}}.$$

The collision cost function is calculated with precomputed sign distance field $sdf(\boldsymbol{x})$ from the original map, where the penalty is added when the $sdf(\boldsymbol{x})$ is smaller than d:

$$\boldsymbol{g}_c(\boldsymbol{\tau}) = \begin{cases} -sdf(\boldsymbol{\tau}) + d & sdf(\boldsymbol{\tau}) \le d \\ 0 & sdf(\boldsymbol{\tau}) > d \end{cases}. \quad (8)$$

While cost guidance enhances collision avoidance rates, it cannot guarantee absolute collision-free operations. Thus, the collision probability gradually diminishes as multiple aerobatic primitives are iteratively generated from preceding ones. To address this, we implement batch sampling followed by an additional coarse collision check step for each generated outcome τ. The coarse collision check module identifies τ_i violating the safety condition ($sdf(\tau_i) < 0$), then iteratively modifies them by replacing colliding trajectories with randomly selected collision-free alternatives in samples. This redundant procedure effectively improves the overall success rate of aerobatic flight.

IV. POST-PROCESSING WITH TRAJECTORY OPTIMIZATION

In this section, we propose to leverage spatial-temporal trajectory optimization to transform the discrete aerobatic primitives into dynamically feasible trajectories. As the section III suggests, the aerobatic diffusion model generates a dense sequence of frames capturing the spatial-attitude dynamics throughout the maneuvers while providing explicit topological information with collision-free properties.

Building on this output, the waypoints presenting key attitude changes and a lightweight safety flight corridor defining free space are extracted, serving as critical inputs and constraints for trajectory optimization. We employ an iterative way to sample sparse waypoints from key frames which present the key flight maneuvers. A key frame is identified when its angular deviation in the body z-axis from its predecessor exceeds a preset threshold α, initialized with

Fig. 4: Five different maneuver styles of aerobatic trajectories: (a) the Power Loop, (b) the Barrel Roll, (c) the Split-S, (d) the Immelmann Turn, (e) the Wall Ride.

the first frame as the reference seed. The corresponding body z-axis z^{ref} along the waypoints serves as the maneuver reference for optimization. For safe corridor generation, we utilize the method in [32] to efficiently generate polyhedrons covering the whole primitive while providing sufficient free space. Additionally, the sequence generated by the diffusion model contains the optimal temporal information learned from the dataset. We directly derive inter-waypoint timestamps from it as the initial time guess for optimization.

With the aforementioned preparations, we construct the trajectory optimization problem as the following formulation:

$$\min_{\boldsymbol{p}(t),\boldsymbol{T}} \mathcal{J} = \mathcal{L}_s + \mathcal{L}_T + \mathcal{L}_{att}, \tag{9}$$

$$s.t.\ \boldsymbol{p}^{(i)}(0) = \boldsymbol{x}_0^{(i)},\ i = 0, ..., s-1, \tag{10}$$

$$\boldsymbol{p}^{(i)}(T) = \boldsymbol{x}_f^{(i)},\ i = 0, ..., s-1, \tag{11}$$

$$\mathcal{G}_\star \leq 0,\ \star = v, f_t, \omega \tag{12}$$

$$\mathcal{G}_{safe} \leq 0, \tag{13}$$

where the MINCO class [33] is used as the trajectory representation, and the waypoints $\boldsymbol{p}(t)$ and time segment $\boldsymbol{T}$ are optimization variables. In the cost function, $\mathcal{L}_s$ and $\mathcal{L}_T$ denote the smooth cost and time cost in normal planning problems. $\mathcal{L}_{att}$ is the cost to align the flight maneuver with the key frame reference as it is expressed as:

$$\mathcal{L}_{att} = \sum_{i=0}^{n} -\cos(\frac{\boldsymbol{f}_t(T_s(i))^\top \boldsymbol{z}_i^{ref}}{\|\boldsymbol{f}_t(T_s(i))\|}), \tag{14}$$

where $T_s(i) = \sum_{j=0}^{i} T_j$, and $\boldsymbol{f}_t$ denotes the net thrust in the world frame, which can be calculated based on differential flatness. Kinodynamic constraints $\mathcal{G}_\star$ are introduced on velocity $\boldsymbol{v}$, net thrust $\boldsymbol{f}_t$ and angular velocity $\boldsymbol{\omega}$, detailed in work [23]. To ensure safety during flight, we constrain each trajectory segment must stay inside the corresponding ith polyhedron:

$$\mathcal{G}_{safe} = \int_0^{T_{sum}} \boldsymbol{A}_i \boldsymbol{p}(t) - \boldsymbol{b}_i\ dt, \tag{15}$$

where $\boldsymbol{A}_i$ and $\boldsymbol{b}_i$ are corresponding parameters of polyhedrons. During the optimization process, we observed that the strong nonlinearity of the z-axis angular velocity can cause the optimization to get trapped in bad suboptimal local minima. This leads to significant violations of angular velocity constraints, which in turn negatively impacts the actual flight performance. To address this issue, we propose a hierarchical optimization strategy that operates in two sequential stages. First, we solve a relaxed problem formulation

by temporarily removing z-axis angular velocity constraints to circumvent local minima. This initial solution then serves as a warm start for the second stage, where we perform a fully constrained optimization refinement that reinstates all dynamic constraints. This simple design improves overall optimization performance while maintaining strict dynamic constraints, thereby ensuring stunning flight performance as detailed in Sec. V-C.

V. RESULTS

In this section, we present a series of experiments to validate the key component designs of our method and evaluate the performance in real-world scenarios. We demonstrate that

1) Explicit conditioning on target points and action semantics enables improved editability of generated aerobatic trajectories.
2) Historical state integration mitigates abrupt transitions between motion primitives while preserving agility.
3) The proposed collision avoidance strategy significantly improves the success rate of aerobatic flight in cluttered environments.
4) The post-processing is essential for bridging discrete planning to dynamically executable trajectories in real-world deployment.

A. Implementation details

Our model is trained on a dataset comprising five distinct aerobatic primitives (Fig. 4), where each primitive is generated by uniformly sampling target waypoints within the spatial bounds of $[0, 8] \times [-6, 6] \times [-1, 1]$ at a resolution of 1.0 meter. To model dynamic transitions between aerobatic primitives, the primitives are generated with supplementary trajectories by randomly sampling waypoints before and after each primitive. The dataset is further augmented through transformations in the global coordinate system, specifically applying discrete z-axis rotations of 90°, 180°, 270° to each of the aerobatic primitives. This symmetry utilization enables omnidirectional maneuver generation while preserving dynamic feasibility constraints, ultimately yielding 450,000 training primitives. The network architecture employs a decoder-only transformer with 4 layers, 4 multi-head attention, and a latent dimension of 256. For the diffusion process, we configure 30 denoising steps with an exponential-noise scheduler. Each primitive sequence spans 6 seconds, discretized into $N_a = 60$ time steps at 0.1 s intervals. To ensure transition continuity, the model incorporates 5-frame historical observations as prior context.

Fig. 5: Up: results of the aerobatic generation conditioned on the "Power Loop" action (a) compared with action-agnostic model (b). Down: box plot of the errors between the terminal of aerobatic primitives and the given target, measured by distances.

B. Simulation Ablations

1) Target and Action Conditions: To validate the effectiveness of target and action conditioning, we compare three model variants: unconstrained generation, target-only conditioning, and target-action joint conditioning. To evaluate target conditioning, 5,000 aerobatic primitives are generated with 50 randomly sampled target waypoints in each of the following four scenarios:

- **In-Distribution Sampling (IDS)**: Targets sampled within the training distribution.
- **Near Out-of-Distribution Sampling (N-OODS)**: Targets sampled outside but near the distribution boundary, defined as $[9, 12] \times [-9, 9] \times [-1, 1]$.
- **Far Out-of-Distribution Sampling (F-OODS)**: Targets sampled far outside the distribution, defined as $[12, 16] \times [-12, 12] \times [-1, 1]$.
- **Unconditional Sampling (UncondS)**: Targets within the distribution but generated by the unconstrained model.

The distribution of distances between primitive terminal positions and target waypoints is visualized as box plots in Fig. 5. Our analysis reveals that the target-only conditioning model reliably guides trajectories to terminate near specified targets, whereas the unconditioned one scatters endpoints randomly due to the absence of target awareness. Surprisingly, the model generalizes to targets near the distribution boundary, demonstrating generalization ability.

To evaluate action conditioning, Fig. 5 visualizes 10 trajectories generated by the target-only and target-action joint conditioning models with the same target. When given "Power Loop" commands, the action-conditioned model produces maneuvers explicitly aligned with semantic intent (Fig. 5(a)), while the action-agnostic model generates inconsistent maneuvers. The above ablation results suggest that target conditioning provides explicit spatial guidance, while action conditioning allows flexible trajectory shaping through

Fig. 6: Comparison between models with and without access to previous primitives. The smoothness is measured with differences of positions δp and Euler angles $\delta\theta$ between adjacent frames.

human-defined commands, enabling controllable generation of task-oriented aerobatic maneuvers.

2) Transition Smoothness: To validate the necessity of historical state conditioning for smooth dynamic transitions, we compare two model variants: with and without access to previous primitive states. Both models are tasked to generate the aerobatic primitives from the same previous trajectory. We quantify motion smoothness by computing adjacent-frame differences in axis-aligned position δp and Euler angles $\delta\theta$. The results are shown in Fig. 6, the model without previous observations failed to understand the dynamic transition process, resulting in abrupt changes in attitude (more than 1 rad) and position (more than 0.5 m) that are infeasible in actual flight. In contrast, the model with observations generates state-coherent maneuvers even during aggressive transitions. Notably, it achieves smooth attitude adjustments and continuous positional updates that align with drone dynamics. This suggests that our model successfully learns the underlying dynamics of drone flight.

3) Collision Avoidance: In this task, we test our method in three different scenarios illustrated in Fig. 7. The factory environment contains small, complex, and unstructured obstacles, while the indoor industrial workshop is extremely narrow with walls obstructing the space, presenting significant challenges for our collision avoidance strategy. $\boldsymbol{p}_t$ are set to be collision-free to guide the aerobatic generation traversing the complex environment and covering the whole flying region. To obtain reliable results, we conduct ablation tests on five different random seeds, where the batch size for sampling in each inference is set to 500. In each ablation, we compare our method with an **UnGuided** baseline (generating samples without cost guidance) and an **UnCheck** baseline (no coarse collision check implemented). The success rate is measured by the proportion of collision-free trajectories among all generated trajectories, where more precise collision checks are performed on both individual motion frames and interpolated trajectories between consecutive frames. For

Fig. 7: The illustration of the drone executing aerobatic maneuvers in three different scenarios: (a) Narrow indoor industrial workshop, (b) Complex outdoor industrial factory, and (c) random forest.

TABLE I: Success rate of different ablation configurations across environments.

	N_{aero}	1(%)	2(%)	3(%)	5(%)	10(%)
	Ours	**99.9 ± 0.1**	**99.8 ± 0.2**	**99.7 ± 0.3**	**99.7 ± 0.3**	**99.4 ± 0.6**
Random Forest	UnGuided	$51.8 ± 8.0$	$7.0 ± 3.0$	$3.1 ± 2.1$	$0.7 ± 0.7$	$0.0 ± 0.0$
	UnCheck	$97.7 ± 2.1$	$85.4 ± 4.2$	$79.9 ± 5.7$	$72.0 ± 6.0$	$53.6 ± 9.4$
	Ours	**100.0 ± 0.0**	**99.9 ± 0.1**	**99.8 ± 0.2**	**99.5 ± 0.5**	**97.2 ± 2.8**
Outdoor Factory	UnGuided	$57.5 ± 3.5$	$9.7 ± 0.9$	$5.8 ± 1.2$	$0.0 ± 0.0$	$0.0 ± 0.0$
	UnCheck	$97.0 ± 1.0$	$82.5 ± 1.7$	$61.9 ± 3.3$	$23.9 ± 3.3$	$7.0 ± 1.6$
	Ours	**99.9 ± 0.1**	**100.0 ± 0.0**	**97.9 ± 2.1**	**98.4 ± 1.6**	**97.1 ± 2.9**
Indoor Workshop	UnGuided	$65.7 ± 18.1$	$43.3 ± 20.3$	$15.4 ± 9.8$	$10.7 ± 8.5$	$0.4 ± 0.4$
	UnCheck	$81.1 ± 12.3$	$62.9 ± 19.9$	$35.6 ± 17.0$	$26. ± 16.5$	$2.6 ± 2.6$

Fig. 8: Snapshot of a quadrotor executing aerobatic flight trajectories with five distinct maneuvers generated by the proposed method in real-world.

Fig. 9: Numerical analysis, δp demonstrates the position error along axis X, Y, Z and total tracking error $\|\delta p\|$. Ref. and Meas. denotes the net thrust values $\|f_t\|$ obtained from planned trajectory and the practical measured value calculated from an inertial measurement unit (IMU) respectively. $\delta\theta$ indicates the rotational angle corresponding to the quaternion that describes the error between desired and actual attitude. ω is the angular velocity along the flight as angular velocity X, Y, Z along the axis and norm $\|\omega\|$.

each baseline, the median success rate and its fluctuation range across all seeds are statistically evaluated as the number of aerobatic maneuvers N_{aero} increases.

Experimental results are summarized in Table I. Our method demonstrates superior success rates across all scenarios compared to two baselines. The comparison reveals that cost guidance contributes most significantly to collision avoidance but cannot guarantee collision-free trajectories in all cases. Specifically, trajectories generated from previously collided motion primitives may compromise subsequent collision-free generation. To address this limitation, the coarse collision check module serves as a lightweight yet effective safeguard, intercepting collision risks in the final trajectory generation process.

C. The Real-World Experiment

To verify the real-world applicability of the proposed method, the aerobatic trajectories containing five aerobatic maneuvers are generated in a narrow and cluttered indoor space with the size of $12 \times 6 \times 4\ m^3$, where a drone executes these trajectories under the NOKOV Motion Capture

System[1]. The real-world performance is demonstrated in Fig. 8, with the numerical analysis provided in Fig. 9. As Fig. 9 indicates, the post-processing constrains the thrust and angular velocity to remain within feasible values, ensuring that the low-level controllers can accurately track the control signals. As a result, the tracking errors in both attitude and position are small, with maximum errors of less than 15 degrees and 0.15 meter respectively. This highlights the critical role of post-processing in practical aerobatic flight generation. Since

[1]https://www.nokov.com/

the discrete outputs of positional and attitude references from the diffusion model would be challenging for the controller to precisely track at the actuator-level commands, they would lead to potential flight failures.

VI. CONCLUSION AND FUTURE WORK

In this work, we unlock the potential of diffusion models for generating long-horizon, multi-maneuver aerobatic trajectories. Our key contribution lies in learning aerobatic primitives with specific conditioning and guidance from trajectory costs, enabling automatic and editable generation. The post-processing further ensures dynamic feasibility, making the method directly deployable on physical drones in the real world. However, the proposed method achieves interaction with the environment primarily through passive obstacle avoidance, which limits the ability to generate truly visually impressive maneuvers that reflect a deep understanding of the environment. Therefore, future work will focus on developing scene-aware aerobatic generation, where trajectories are dynamically crafted in response to environmental features (e.g., flips through narrow gaps). With a better understanding of the environments, we believe that it can create more visually appealing maneuvers, blending agility with surroundings in a way that enhances both performance and aesthetic value.

REFERENCES

[1] D. Tezza, D. Caprio, D. Laesker, and M. Andujar, "Let's fly! an analysis of flying fpv drones through an online survey.," in *iHDI@ CHI*, 2020.

[2] E. Kaufmann, A. Loquercio, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza, "Deep drone acrobatics," *ArXiv*, vol. abs/2006.05768, 2020.

[3] G. Lu, Y. Cai, N. Chen, F. Kong, Y. Ren, and F. Zhang, "Trajectory generation and tracking control for aggressive tail-sitter flights," *The International Journal of Robotics Research*, vol. 43, pp. 241 – 280, 2022.

[4] G. Lu, W. Xu, and F. Zhang, "On-manifold model predictive control for trajectory tracking on robotic systems," *IEEE Transactions on Industrial Electronics*, vol. 70, pp. 9192–9202, 2023.

[5] E. Tal, G. Ryou, and S. Karaman, "Aerobatic trajectory generation for a vtol fixed-wing aircraft using differential flatness," *IEEE Transactions on Robotics*, vol. 39, pp. 4805–4819, 2022.

[6] A. Serifi, E. Zürich, S. D. Research, D. R. S. E. K. R. GRANDIA, E. Z. S. M. GROSS, and S. M. Bächer, "Robot motion diffusion model: Motion generation for robotic characters," in *ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia*, 2024.

[7] H. Yi, J. Thies, M. J. Black, X. B. Peng, and D. Rempe, "Generating human interaction motions in scenes with text control," *ArXiv*, vol. abs/2404.10685, 2024.

[8] S. Cohan, G. Tevet, D. Reda, X. B. Peng, and M. van de Panne, "Flexible motion in-betweening with diffusion models," in *International Conference on Computer Graphics and Interactive Techniques*, 2024.

[9] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *International Conference on Machine Learning*, 2022.

[10] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters, "Motion planning diffusion: Learning and planning of robot motions with diffusion models," *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1916–1923, 2023.

[11] S. Huang, Z. Wang, P. Li, B. Jia, T. Liu, Y. Zhu, W. Liang, and S.-C. Zhu, "Diffusion-based generation, optimization, and planning in 3d scenes," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16750–16761, 2023.

[12] J.-H. Tseng, R. Castellon, and C. K. Liu, "Edge: Editable dance generation from music," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 448–458, 2022.

[13] U. A. Mishra, S. Xue, Y. Chen, and D. Xu, "Generative skill chaining: Long-horizon skill planning with diffusion models," in *Conference on Robot Learning*, 2023.

[14] S. Yang, Z. Yang, and Z. Wang, "Longdancediff: Long-term dance generation with conditional diffusion model," *ArXiv*, vol. abs/2308.11945, 2023.

[15] G. Tevet, S. Raab, S. Cohan, D. Reda, Z. Luo, X. B. Peng, A. Bermano, and M. van de Panne, "Closd: Closing the loop between simulation and diffusion for multi-task character control," *ArXiv*, vol. abs/2410.03441, 2024.

[16] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. V. Gool, "Repaint: Inpainting using denoising diffusion probabilistic models," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461, 2022.

[17] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *ArXiv*, vol. abs/2105.05233, 2021.

[18] Y. Song, J. N. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *ArXiv*, vol. abs/2011.13456, 2020.

[19] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, 2024.

[20] A. K. Sridhar, D. Shah, C. Glossop, and S. Levine, "Nomad: Goal masked diffusion policies for navigation and exploration," *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 63–70, 2023.

[21] D. Mellinger and V. R. Kumar, "Minimum snap trajectory generation and control for quadrotors," *2011 IEEE International Conference on Robotics and Automation*, pp. 2520–2525, 2011.

[22] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robotics and Automation Letters*, vol. 3, pp. 620–626, 2017.

[23] Z. Wang, C. Xu, and F. Gao, "Robust trajectory planning for spatial-temporal multi-drone coordination in large scenes," *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12182–12188, 2021.

[24] Y. Chen and N. O. Pérez-Arancibia, "Controller synthesis and performance optimization for aerobatic quadrotor flight," *IEEE Transactions on Control Systems Technology*, vol. 28, pp. 2204–2219, 2020.

[25] S. Lupashin, A. P. Schoellig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," *2010 IEEE International Conference on Robotics and Automation*, pp. 1642–1648, 2010.

[26] X. Zhou, X. Wen, Z. Wang, Y. Gao, H. Li, Q. Wang, T. Yang, H. Lu, Y. Cao, C. Xu, and F. Gao, "Swarm of micro flying robots in the wild," *Science Robotics*, vol. 7, 2022.

[27] B. Zhou, H. Xu, and S. Shen, "Racer: Rapid collaborative exploration with a decentralized multi-uav system," *IEEE Transactions on Robotics*, vol. 39, pp. 1816–1835, 2022.

[28] Y. Gao, J. Ji, Q. Wang, R. Jin, Y. Lin, Z. Shang, Y. Cao, S. Shen, C. Xu, and F. Gao, "Adaptive tracking and perching for quadrotor in dynamic scenarios," *IEEE Transactions on Robotics*, vol. 40, pp. 499–519, 2023.

[29] Z. Zhang, Y. Zhong, J. Guo, Q. Wang, C. Xu, and F. Gao, "Auto filmer: Autonomous aerial videography under human interaction," *IEEE Robotics and Automation Letters*, vol. 8, pp. 784–791, 2023.

[30] G. Tevet, S. Raab, B. Gordon, Y. Shafir, D. Cohen-Or, and A. H. Bermano, "Human motion diffusion model," *ArXiv*, vol. abs/2209.14916, 2022.

[31] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5738–5746, 2018.

[32] Q. Wang, Z. Wang, C. Xu, and F. Gao, "Fast iterative region inflation for computing large 2-d/3-d convex regions of obstacle-free space," *ArXiv*, vol. abs/2403.02977, 2024.

[33] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," *IEEE Transactions on Robotics*, vol. 38, pp. 3259–3278, 2021.