

这是一个有英文单词 RapidIO 的论文标题

戴英特

2026 年 6 月

中图分类号： TQ028.1

UDC分类号： 540

这是一个有英文单词 RapidIO 的论文标题

作者姓名 戴英特

学院名称 自动化学院

指导教师 甘明刚教授

答辩委员会主席 ** 教授

申请学位 工学硕士

学科专业 控制工程

学位授予单位 北京理工大学

论文答辩日期 2026 年 6 月

A Thesis Title with English word RapidIO

Candidate Name: Daiyingte

School or Department: School of Information and Electronics

Faculty Mentor: Prof. **

Chair, Thesis Committee: Prof. **

Degree Applied: Master

Major: Electronics Science and Technology

Degree by: Beijing Insititute of Technology

The Date of Defence: 6, 2019

这是 | 一个有关英文单页RapidIO的竖排标题

戴英特

北京理工大学

研究成果声明

本人郑重声明：所提交的学位论文是我本人在指导教师的指导下进行的研究工作获得的研究成果。尽我所知，文中除特别标注和致谢的地方外，学位论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京理工大学或其它教育机构的学位或证书所使用过的材料。与我一同工作的合作者对此研究工作所做的任何贡献均已在学位论文中作了明确的说明并表示了谢意。

特此申明。

作者签名：_____ 签字日期：_____

关于学位论文使用权的说明

本人完全了解北京理工大学有关保管、使用学位论文的规定，其中包括：①学校有权保管、并向有关部门送交学位论文的原件与复印件；②学校可以采用影印、缩印或其它复制手段复制并保存学位论文；③学校可允许学位论文被查阅或借阅；④学校可以学术交流为目的，复制赠送和交换学位论文；⑤学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

作者签名：_____ 导师签名：_____

签字日期：_____ 签字日期：_____

摘要

四旋翼无人机在高速密集障碍环境中的自主避障是机器人领域的关键挑战。传统模块化导航系统在高速条件下面临流水线延迟累积与误差跨模块传播的固有瓶颈，端到端学习控制方法通过将高维视觉观测直接映射为控制指令，为突破上述瓶颈提供了新的技术路径。然而，端到端方法在高速闭环部署中仍面临时序建模能力不足、流式推理状态管理脆弱以及安全性与平滑性冲突等系统性问题。

本文面向高速端到端视觉避障任务，提出以 ViT 空间编码与 Mamba 选择性状态空间模型时序聚合为核心的策略网络架构，构建了涵盖训练方法（BC+DAgger 闭环数据增强）、部署约束（RACS 动态速率限制）与评测协议的完整系统。主要工作与贡献包括以下三个方面：

第一，提出 ViT+Mamba 端到端策略网络并建立多速度档系统评测体系。在 Flight-mare 仿真平台中，以行为克隆为基础训练范式，在 5 个速度档与同分布/分布外双环境下进行系统评测。结果表明，Mamba 的选择性时序聚合能力使 ViT+Mamba 在高速段的碰撞率与碰撞事件次数显著优于 ViT+LSTM 基线，且分布外泛化优势同样显著。在此基础上，引入 DAgger 闭环数据增强（3 轮迭代），在强 BC 基线之上进一步降低高速段碰撞频次与碰撞持续时间，跨试验行为方差显著收敛，工程部署稳定性大幅提升。同时，设计部署侧 RACS 动态速率限制模块，以低于 0.1ms 的计算开销实现 Command Jerk 的显著降低，安全性基本保持。

第二，揭示序列模型在端到端控制流式部署中的一个关键陷阱：状态管理错误导致的无记忆退化。通过系统对比实验发现，当序列模型的内部状态在每个推理步被错误重置时，碰撞率从 0% 飙升至 90%，Mean Y Drift 从 0.022m 增至 0.770m——这一毁灭性后果此前在端到端控制文献中缺乏系统性报道。本文提出回合边界级状态生命周期管理协议与硬防护机制（运行时断言、配置锁定、可审计日志），确保部署一致性与评测结论的可信性。

第三，探索从混合架构（ViT+Mamba）走向全 SSM 架构（MambaVision+Mamba）的可行性。在保持时序模块与训练流程完全不变的条件下，将视觉编码器替换为 MambaVision，形成空间--时间统一的 SSM 系列架构，为理解 SSM 在视觉--运动控制任务中的能力边界提供实证基础。

关键词：端到端视觉避障；选择性状态空间模型；Mamba；流式部署一致性；行为克隆；DAgger

Abstract

Autonomous obstacle avoidance for quadrotor UAVs in high-speed, densely cluttered environments is a critical challenge in robotics. Traditional modular navigation systems suffer from inherent bottlenecks of pipeline latency accumulation and cross-module error propagation under high-speed conditions. End-to-end learning-based control, which directly maps high-dimensional visual observations to control commands, offers a promising alternative. However, such methods still face systematic issues in high-speed closed-loop deployment, including insufficient temporal modeling capability, fragile streaming inference state management, and conflicts between safety and smoothness.

This thesis addresses the high-speed end-to-end visual obstacle avoidance task by proposing a policy network architecture centered on ViT spatial encoding and Mamba selective state space model temporal aggregation, and constructs a complete system encompassing training methods (BC + DAgger closed-loop data augmentation), deployment constraints (RACS dynamic rate limiting), and evaluation protocols. The main contributions are as follows:

First, the ViT+Mamba end-to-end policy network is proposed with a multi-speed systematic evaluation framework. Using behavioral cloning as the base training paradigm in the Flightmare simulation platform, systematic evaluation is conducted across five speed tiers and both in-distribution (Spheres) and out-of-distribution (Trees) environments. Results demonstrate that Mamba's selective temporal aggregation capability yields significantly lower collision rates and collision counts compared to the ViT+LSTM baseline at high speeds, with the out-of-distribution generalization advantage being equally pronounced. Building upon the strong BC baseline, DAgger closed-loop data augmentation (3 iterations) further reduces collision frequency and duration at high speeds, with cross-trial behavioral variance converging significantly. Additionally, the deployment-side RACS dynamic rate limiter achieves substantial Command Jerk reduction with less than 0.1ms computational overhead while maintaining safety.

Second, a critical pitfall in streaming deployment of sequence models for end-to-end control is revealed: erroneous state management leading to memoryless degradation. Through systematic ablation experiments, it is found that when the internal states of sequence models

are incorrectly reset at every inference step, the collision rate surges from 0% to 90%, and Mean Y Drift increases from 0.022m to 0.770m—a devastating consequence that has lacked systematic reporting in the end-to-end control literature. An episode-boundary state lifecycle management protocol with hard safeguards (runtime assertions, configuration locking, and auditable logging) is proposed to ensure deployment consistency and evaluation credibility.

Third, the feasibility of transitioning from a hybrid architecture (ViT+Mamba) to a fully SSM-based architecture (MambaVision+Mamba) is explored. By replacing the visual encoder with MambaVision while keeping the temporal module and training pipeline unchanged, a spatially-temporally unified SSM architecture is formed, providing empirical evidence for understanding the capability boundaries of SSMs in visual-motor control tasks.

Key Words: End-to-end visual obstacle avoidance; Selective state space model; Mamba; Streaming deployment consistency; Behavioral cloning; DAgger

主要符号对照表

BIT	北京理工大学的英文缩写
L <small>A</small> T <small>E</small> X	一个很棒的排版系统
L <small>A</small> T <small>E</small> X 2 ϵ	一个很棒的排版系统的最新稳定版
X <small>E</small> T <small>E</small> X	L <small>A</small> T <small>E</small> X 的好兄弟，事实上他有很多个兄弟，但是这个兄弟对各种语言的支持能力都很强
cTEX	成套的中文 L <small>A</small> T <small>E</small> X 解决方案，由一帮天才们开发
H ₂ SO ₄	硫酸
$e^{\pi i} + 1 = 0$	一个集自然界五大常数一体的炫酷方程
2H ₂ + O ₂ ——> 2H ₂ O	一个昂贵的生成生命之源的方程式

目 录

摘要	I
Abstract	III
主要符号对照表	V
第 1 章 绪论	1
1.1 研究背景与问题提出	1
1.2 研究意义与应用价值	2
1.2.1 工程与应用意义	2
1.2.2 学术意义：从“网络结构”走向“部署一致性与可审计”	3
1.3 高速端到端视觉避障的关键挑战	3
1.3.1 高速闭环对延迟极度敏感	3
1.3.2 密集环境下的观测不确定性	4
1.3.3 时序建模与流式部署一致性	4
1.3.4 安全性与平滑性的冲突	4
1.3.5 有限算力与实时性约束	5
1.3.6 闭环分布偏移与训练数据局限	5
1.4 研究内容与技术路线	6
1.4.1 总体研究目标	6
1.4.2 技术路线概述	6
1.5 本文主要贡献与创新点	8
1.6 论文结构安排	9
第 2 章 相关工作与国内外研究现状	10
2.1 模块化自主飞行：感知--规划--控制范式	10
2.1.1 视觉/视觉惯性里程计与 SLAM	10
2.1.2 路径规划与轨迹规划	11
2.1.3 高保真仿真平台与可复现实验基座	12

2.2 端到端视觉飞行控制：从轻量网络到高速野外与冠军竞速	13
2.2.1 模仿学习、特权信息专家与 Sim-to-Real	13
2.2.2 强化学习与竞速：从“可飞”到“超越人类冠军”	13
2.2.3 模仿学习方法谱系与闭环分布偏移问题	14
2.3 时序建模：RNN/LSTM、Transformer 与结构化状态空间模型	14
2.3.1 RNN/LSTM 的流式优势与局限	15
2.3.2 Transformer 与视觉 Transformer (ViT)	15
2.3.3 结构化状态空间模型：S4 与 Mamba	16
2.4 视觉 Backbone 演进：CNN→ViT→MambaVision	17
2.4.1 卷积神经网络 (CNN)	17
2.4.2 视觉 Transformer (ViT) 的表征优势	17
2.4.3 MambaVision：混合 Mamba-Transformer 视觉骨干	18
2.5 安全性、平滑性与部署侧约束机制	18
2.5.1 安全学习控制研究概述	18
2.5.2 控制障碍函数与 MPC 证书	19
2.5.3 部署侧平滑与速率限制	19
2.6 国内外研究现状总结与本文切入点	20
2.6.1 国外研究趋势	20
2.6.2 国内研究现状	21
2.6.3 研究趋势与本文切入点	22
第 3 章 问题定义与系统框架	23
3.1 任务定义与回合终止条件	23
3.1.1 任务形式化	23
3.1.2 评测环境	23
3.1.3 回合终止条件	24
3.2 观测空间与动作空间	24
3.2.1 深度图像观测	24
3.2.2 轻量状态输入	24

3.2.3 动作空间	25
3.3 闭环控制回路与部署形态	25
3.3.1 系统架构	25
3.3.2 仿真平台	26
3.3.3 控制频率与延迟预算	26
3.4 特权信息专家与数据生成	26
3.4.1 专家策略设计	26
3.4.2 训练数据集	27
3.5 评价指标与统计协议	27
3.5.1 安全性指标	27
3.5.2 平滑性指标	28
3.5.3 系统性能指标	28
3.5.4 统计方式	29
3.6 评测可审计规范	29
第 4 章 ViT+Mamba 策略网络与训练方法	30
4.1 总体架构	30
4.1.1 三层系统架构	30
4.2 视觉编码器：轻量化 ViT	30
4.2.1 架构设计	31
4.3 时序聚合模块：Temporal Mamba	31
4.3.1 选择性状态空间模型	31
4.3.2 模块配置	32
4.3.3 Batch 训练与 Streaming 推理的双模式接口	32
4.4 特征融合与控制头	33
4.5 训练目标与损失函数	33
4.5.1 行为克隆监督损失	33
4.5.2 指令抖动惩罚 (Jerk Loss)	34
4.5.3 三阶段课程学习策略	34

4.6	训练实现细节	34
4.7	基线设置与公平对比	34
4.7.1	ViT+LSTM 基线	34
4.7.2	方法对比列表	35
4.8	DAgger 闭环数据增强：方法与实现	36
4.8.1	动机：闭环分布偏移与 DAgger 框架	36
4.8.2	算法与执行策略	36
4.8.3	评测指标审计与重算规范	37
4.9	部署侧安全约束：RACS 动态速率限制	38
4.9.1	问题动机与设计目标	38
4.9.2	算法定义与数学形式	39
4.9.3	在安全学习方法谱系中的定位	40
4.9.4	权衡分析与扩展路径	41
第 5 章	流式部署一致性与状态生命周期管理	42
5.1	训练与部署的语义差异	42
5.1.1	Batch 训练模式	42
5.1.2	Streaming 推理模式	42
5.1.3	两种模式的等价性条件	43
5.2	错误状态重置导致无记忆退化：机理分析	43
5.2.1	问题描述	43
5.2.2	退化机理	43
5.2.3	闭环后果	43
5.2.4	问题的隐蔽性	44
5.3	回合边界状态生命周期协议	44
5.3.1	核心原则	44
5.3.2	实现细节	45

5.4 硬防护机制与可审计日志	45
5.4.1 运行时断言	45
5.4.2 配置锁定	46
5.4.3 可审计日志	46
5.5 实验验证: KeepState 与 ResetState 对比	46
5.5.1 定量结果	46
5.5.2 漂移可视化	47
5.6 Batch–Streaming 等价性验证	48
5.7 普适性讨论	49
5.7.1 不同序列模型的影响	49
5.7.2 贡献定位	49
5.7.3 对评测可信度的启示	49
第 6 章 实验设置与结果分析	51
6.1 实验平台与环境配置	51
6.1.1 仿真平台	51
6.1.2 障碍环境	51
6.2 评测协议	51
6.3 主结果: ViT+Mamba 与 ViT+LSTM 的系统对比	52
6.3.1 碰撞率与碰撞事件次数	52
6.4 分布外泛化分析: Trees 环境	53
6.5 消融实验	54
6.5.1 时序模块消融: Mamba vs LSTM	54
6.5.2 Jerk Loss 消融	54
6.5.3 状态管理消融: KeepState vs ResetState	54
6.6 推理延迟与系统时序分析	55
6.6.1 单步推理时间	55
6.6.2 控制周期分布	55

6.7 失败案例分析	55
6.7.1 高速窄通道碰撞	55
6.7.2 OOD 环境下的误判模式	56
6.7.3 时序记忆衰减与恢复延迟	57
6.8 RACS 部署侧约束实验	58
6.8.1 实验设置	58
6.8.2 实验结果	58
6.8.3 RACS 实验小结	58
6.9 DAgger 闭环数据增强实验	59
6.9.1 实验设置	59
6.9.2 主要结果：碰撞事件次数与碰撞持续时间随轮次下降	60
6.9.3 行为一致性：方差收敛是 DAgger 的主要收益	62
6.9.4 分布可视化	63
6.9.5 BC 与 DAgger-final 的对比：同分布与分布外性能	64
6.9.6 DAgger 实验小结	65
6.10 从混合架构到全 SSM 架构：MambaVision 探索实验	66
6.10.1 实验设计	66
6.10.2 评测维度	67
6.10.3 预期假设与理论动机	67
结论	68
参考文献	71
附录 A ***	74
附录 B Maxwell Equations	75
攻读学位期间发表论文与研究成果清单	76
致谢	77
作者简介	78

插 图

图 1.1 本文技术路线总览	7
图 3.1 端到端闭环控制回路时序示意	25
图 4.1 端到端导航管线架构总览	30
图 4.2 不同方法的 Command Jerk 随速度变化趋势。ViT+Mamba 的 jerk 在中高速段整体高于 ViT+LSTM 基线，反映了更敏捷时序响应的代价。	39
图 5.1 流式推理中 KeepState 与 ResetState 的漂移对比。ResetState（逐步重置）导致显著的横向漂移趋势，反映出时序模型在无记忆退化下的闭环不稳定行为。	47
图 5.2 KeepState 与 ResetState 模式下横向漂移的累积对比	48
图 6.1 ViT+Mamba 与 ViT+LSTM 基线在 Spheres/Trees 环境下的性能对比。(a) 全程碰撞率随速度变化；(b) 碰撞事件次数随速度变化；(c) Command Jerk 随速度变化；(d) 单步推理时间随速度变化。	52
图 6.2 控制循环周期 Δt 的分布统计。该分布用于检验不同方法的系统时序一致性，并排除负载差异造成的混淆因素。	56
图 6.3 RACS 在 Spheres 环境下的验证结果。对比 Mamba No RACS、Mamba + RACS (Ours) 以及 ViT+LSTM 基线在不同速度下的安全性与平滑性表现。	59
图 6.4 碰撞事件次数随 DAgger 轮次变化 (Spheres 环境, 9 m/s 与 12 m/s)。R0 为 BC 基线，R1-R3 为 DAgger 各轮次。误差条表示 10 次试验的标准差。	61
图 6.5 平均碰撞持续时间随 DAgger 轮次变化 (Spheres 环境, 9 m/s 与 12 m/s)。碰撞持续时间采用“含零”口径（见第 4 章 4.8.3 节说明）。	61
图 6.6 碰撞指标跨试验标准差随 DAgger 轮次的演化。左：Collision Count std；中：Collision Duration std；右：Collision Rate std。标准差下降反映了策略行为从“有时好有时差”收敛为“稳定表现”。注意：Duration std 采用含零口径（与图 6.5 一致），因此其趋势可能与条件分布口径（图 6.8）有所不同。	62
图 6.7 碰撞事件次数的逐轮分布。可以观察到分布随轮次逐步集中于低值区域。	63

图 6.8 碰撞持续时间的逐轮分布（条件分布，仅统计 duration > 0 的回合）。注意该图采用条件分布口径，主文结论采用“含零”口径，两者差异见第 4 章 4.8.3 节说明。	64
图 6.9 BC 基线与 DAgger-R3 在 Spheres (ID) 与 Trees (OOD) 环境下的碰撞率对比（各速度档）。	64
图 6.10 Trees OOD 环境下 BC 与 DAgger-R3 的多维指标对比（Command Jerk、完成时间、平均前向速度）。	65

表 格

表 2.1 高速端到端视觉避障相关技术路线对比	20
表 4.1 ViT 编码器各阶段张量尺寸	31
表 4.2 Temporal Mamba 模块配置	32
表 4.3 策略网络训练超参数配置	35
表 4.4 DAgger 迭代执行配置	37
表 5.1 流式状态管理消融实验（KeepState vs ResetState）	47
表 6.1 评测协议核心参数	52
表 6.2 Spheres 环境主结果：碰撞率与碰撞事件次数（均值）	53
表 6.3 Trees 环境主结果：碰撞率与碰撞事件次数（均值）	53
表 6.4 消融实验汇总	55

第 1 章 绪论

1.1 研究背景与问题提出

四旋翼无人机凭借高机动性、垂直起降与悬停能力，在巡检、搜索救援、环境监测、应急通信以及室内外自主作业等任务中具有广泛应用前景。然而，当飞行任务从“低速、开阔、静态”逐步走向“高速、密集、动态”的复杂场景时，自主飞行面临的核心矛盾会显著加剧：一方面，高速会放大传感噪声、执行延迟与建模误差在闭环中的累积效应；另一方面，密集障碍环境要求系统在极短时间内完成感知、决策与控制，并在强不确定性下保持鲁棒性。Loquercio 等在 Learning High-Speed Flight in the Wild 中明确指出：传统将导航拆分为感知、建图、规划等子模块的做法在低速时效果较好，但在高速密集环境中会因为流水线式延迟与误差传递而变得脆弱；他们提出端到端映射以降低延迟、提升鲁棒性，并展示了在复杂真实环境中的高速飞行能力 [1]。

在机器人与无人机自主飞行领域，主流方案长期采用模块化范式（Perception–Planning–Control），并通过视觉/视觉惯性里程计、SLAM、地图构建、局部/全局规划和低层控制器来实现闭环导航。该范式的优势在于工程可解释性强、模块边界清晰、便于调参与验证。ORB-SLAM2[2] 与 VINS-Mono[3] 分别代表了稀疏特征 SLAM 与视觉惯性紧耦合估计的代表性工作，为状态估计提供了高精度基础设施。在规划层面，RRT* 与 PRM* 给出了渐近最优采样规划的理论基础 [4]；FASTER 则提出同时维护快速轨迹与安全回退轨迹以支持更高速度上限 [5]。然而，模块化方案的潜在代价是：系统延迟随模块串联增加、误差跨模块传播、以及模块间假设不一致。这些问题在高速飞行时尤其突出：串联推理延迟等效为状态预测误差，感知误差、建图误差与规划误差的复合传播最终导致避障失败或轨迹振荡。

与此同时，端到端学习控制逐渐成为高速飞行的一条重要路径。端到端方法通过将高维观测直接映射为控制量或短期轨迹，避免显式建图与复杂规划带来的计算与时延瓶颈，并可在训练中吸收大量仿真数据，以特权信息专家生成示范来提升安全性与泛化。端到端控制的思想可追溯到 Pomerleau 提出的 ALVINN[6]，其将神经网络直接用于自动驾驶车道保持。NVIDIA 的端到端自动驾驶系统进一步验证了深度卷积网络从摄像头图像直接回归转向角的可行性 [7]。在无人机领域，DroNet 将视觉输入映射为转向与碰撞概率，实现城市环境中的端到端导航 [8]；CAD2RL 通过在纯合成环境

中训练并迁移到真实室内场景，展示了仿真到现实迁移的潜力 [9]；Gandhi 等则提出通过大量碰撞数据进行自监督学习以获取避障能力 [10]。Deep Drone Racing 进一步利用域随机化实现从仿真到真实竞速环境的零样本迁移 [11]。

近年来，强化学习也在竞速场景推动了端到端系统能力上限。Kaufmann 等提出的 Swift 系统结合仿真深度强化学习与真实数据校正，在真实对抗竞速中达到了与人类冠军同级甚至胜出的水平 [12]，代表了端到端方法在极限工况下的里程碑式进展。该成果表明，在充分的仿真基础设施、数据闭环与系统化工程实现支撑下，端到端系统不仅可以在简单场景替代传统流水线，更能在极端动态条件下展现出超越人类操控的性能上限。

总结而言，高速端到端视觉避障的价值不仅在于“替代模块化”，更在于以更短时延、更强时序建模能力支撑闭环稳定性。而当系统部署在流式推理（Streaming Inference）的在线控制回路中时，“时间建模 + 状态一致性 + 工程可复现”会成为决定性能上限的关键因素。如何在保持端到端方法低延迟优势的同时，解决其在部署可信性、安全约束与可复现评测方面的不足，构成了本文的核心研究动机。

1.2 研究意义与应用价值

1.2.1 工程与应用意义

高速避障能力直接决定无人机在复杂场景中的可用性。例如：林区穿越、坍塌建筑侦察、狭窄空间巡检等任务普遍存在密集障碍和不可预知扰动；若系统只能在低速下安全飞行，则任务效率与覆盖能力会受到严重限制。端到端方法通过减少显式地图与规划计算，使得在有限算力平台上实现更高刷新率的闭环控制成为可能。

具体而言，工程意义体现在以下方面。首先，传统模块化系统在机载嵌入式平台上往往需要同时运行 SLAM、规划器与控制器，三者的算力分配与调度本身就是工程难题；端到端方法将感知到控制压缩为单次神经网络前向推理，显著简化了系统架构与部署复杂度。其次，在灾后搜救、林区巡检等时间敏感场景中，飞行速度直接关联任务效率：以 3 m/s 与 10 m/s 的速度对比，同一任务覆盖面积可相差三倍以上。因此，在安全前提下提升飞行速度具有直接的任务价值。最后，端到端框架的模块化程度更低，使得算法迭代与仿真--现实迁移的周期更短，有利于快速原型验证与工程闭环。

1.2.2 学术意义：从“网络结构”走向“部署一致性与可审计”

端到端控制研究中常见的风险是：论文所报告的性能指标可能被工程实现细节所污染。尤其是涉及序列模型时，训练（Batch 序列前向）与推理（Streaming 单步递推）模式不一致会导致“看似提升/退化”的假象。当策略包含 LSTM[13]、Transformer[14] 或结构化状态空间模型 [15] 等序列模型，并以流式方式部署时，训练与部署之间的状态管理差异会显著影响行为一致性：若工程实现中误将内部状态在每个时间步或每次推理调用时重置，序列模型将退化为“无记忆策略”，丧失时序聚合能力，进而引发系统性漂移并污染实验结论。

这一问题在当前端到端控制文献中缺乏系统性讨论。本文将流式部署一致性作为独立贡献进行分析，不仅给出现象与成因的系统描述，还提出回合边界级状态生命周期管理与硬防护机制，并建立可审计的评测协议。这使得本文的贡献从“提出一个新的网络结构”提升到“提出可复现、可审计的部署一致性方法论”——在硕士论文层面，这一维度的工程严谨性具有独立的学术价值。

此外，本文探索将结构化状态空间模型从时序建模进一步拓展到空间编码：通过引入 MambaVision[16] 作为视觉 backbone，与时序 Mamba[15] 形成“空间--时间统一的 SSM 系列架构”，为端到端视觉控制系统的表征效率与架构一致性提供新的设计思路与实验证据。

1.3 高速端到端视觉避障的关键挑战

结合已有研究与工程实践，高速端到端视觉避障通常面临以下五项关键挑战：

1.3.1 高速闭环对延迟极度敏感

在高速飞行中，感知噪声、执行延迟与动力学不确定性会通过闭环耦合被显著放大。以 10 m/s 的飞行速度为例，50 ms 的额外延迟即意味着 0.5 m 的位置预测偏差——在密集障碍环境中，这一偏差足以导致碰撞。模块化系统中，感知--规划--控制的串联推理延迟会等效为状态预测误差，导致控制滞后、避障反应不及时与安全裕度降低。端到端策略虽可减少流水线延迟，但仍需在噪声观测条件下做出稳定可靠决策，并在高速下保持闭环稳定 [1]。因此，如何在有限算力下实现低延迟且鲁棒的闭环控制，是高速端到端飞行的首要挑战。

1.3.2 密集环境下的观测不确定性

快速运动带来的运动模糊、深度噪声、遮挡与纹理缺失会严重降低几何估计的可靠性。在低速条件下，传感误差通常可以被状态估计的滤波或平滑机制有效抑制；但在高速条件下，观测频率相对于运动变化率的比值下降，每帧图像的信息量变低，且相邻帧之间的视觉外观变化剧烈。端到端策略必须对这些不确定性具备内在鲁棒性——不仅依赖训练数据分布的覆盖，还需要在架构层面通过时序聚合来抑制单帧噪声的影响。

1.3.3 时序建模与流式部署一致性

高速避障并非静态映射问题：策略必须利用短时历史信息来抑制观测噪声、捕捉障碍相对运动趋势并稳定控制输出。传统做法多使用 LSTM/RNN[13] 进行时序聚合，但可能面临长序列训练稳定性、计算瓶颈以及部署状态管理敏感等问题。结构化状态空间模型（SSM）提供了另一条路径：例如 Mamba 提出选择性状态空间模型，强调线性复杂度与高吞吐的序列建模能力 [15]，为在线控制中的时序建模提供潜在优势。

然而，更深层的挑战在于流式推理一致性。序列模型在线推理时依赖内部状态持续传播：每个控制周期输入当前观测并更新内部状态。训练与部署的模式差异会带来严重的一致性风险——训练往往采用定长序列 batch 前向，部署则以单步递推更新。一旦状态在错误时刻被重置（例如每次推理调用时重新初始化），模型会退化为“无记忆策略”，进而触发系统性漂移与性能崩坏。这类问题往往不易在离线评测中暴露，但会在真实闭环里被放大。因此，必须通过严格的状态生命周期管理与硬防护机制加以解决。

1.3.4 安全性与平滑性的冲突

更敏捷的策略往往能够减少碰撞率，但也可能产生更高频率的控制指令抖动（command jerk），影响执行器寿命、能耗与飞行平滑性。安全与平滑之间的张力是一个内在矛盾：更激进的避障动作意味着更大幅度和更高频率的控制量变化，而过度平滑又可能导致避障不及时。

安全学习领域已提出多种路线。Brunke 等对安全学习控制进行了系统综述，总结了训练侧约束、运行时证书与安全滤波等主要方法类别 [17]。基于控制障碍函数

(CBF) 的安全强化学习框架可在学习控制中强制满足安全约束 [18]; MPSC (model predictive safety certification) 则通过 MPC 可行性证书对学习控制输出进行最小修改以满足约束 [19]。对于高速端到端避障系统，在保证安全性的前提下降低 jerk 并建立可部署的平滑机制，是工程落地的重要环节。训练侧约束、部署侧速率限制或安全滤波，以及安全证书模块均是候选方案，需要根据具体系统特性进行权衡选择。

1.3.5 有限算力与实时性约束

端到端策略要在真实系统中落地，通常受限于机载算力、控制周期和推理延迟。以典型的机载计算平台（如 NVIDIA Jetson 系列）为例，GPU 算力与桌面级设备存在数量级差距；而控制回路通常要求 20Hz 至 50Hz 的刷新率，对应每次推理的时间预算仅为 20ms 至 50ms。这一约束直接限制了策略网络的复杂度上限。

在视觉 backbone 方面，基于自注意力的 ViT[20] 在表征能力上具有优势，但其二次方复杂度在高分辨率输入下可能成为瓶颈。Mamba[15] 的线性复杂度使其在序列建模中更具部署友好性。近期 MambaVision[16] 将 Mamba 思想引入视觉 backbone 设计，在保持高表征能力的同时实现更优的效率--精度权衡。高效 backbone 与线性复杂度的序列建模结构因此对机载部署更具吸引力。

1.3.6 闭环分布偏移与训练数据局限

上述五项挑战均涉及系统层面的设计决策，而从学习算法角度审视，端到端避障还面临一个根本性的**分布偏移**（Distribution Shift / Covariate Shift）问题 [21]。

行为克隆（BC）是端到端控制中最常用的训练范式：以专家策略生成的状态--动作对为监督信号，通过最小化策略输出与专家动作之间的损失进行离线学习。然而，BC 的训练数据由**专家策略**诱导的状态分布生成，而实际部署时策略访问的状态分布由**学生策略自身**诱导。当学生策略在某些状态下产生微小偏差时，后续状态会偏离专家数据的覆盖范围，导致预测误差累积——这就是经典的“误差复合”（compounding error）现象 [21]。

在高速避障场景中，分布偏移的代价尤为严重：

- 高速下策略的微小偏差会在极短时间内放大为显著的轨迹偏移，使无人机进入训练数据从未覆盖的状态区域；

- 专家数据通常在“正常飞行”条件下采集，对“接近碰撞”与“碰撞后恢复”等边界状态的覆盖天然不足；
- 即使 BC 基线在均值层面表现良好，跨试验的行为方差可能较大——策略在部分试验中表现优异，在另一些试验中因进入未覆盖状态区域而表现显著退化。

DAgger (Dataset Aggregation) [21] 通过在线采集当前策略诱导的闭环数据并由专家标注，逐步缩小训练分布与部署分布之间的差距，为缓解 BC 的分布偏移问题提供了理论与实践基础。本文在第 4 章将 DAgger 引入 ViT+Mamba 系统，并在第 6 章给出实验验证。

1.4 研究内容与技术路线

1.4.1 总体研究目标

本文面向高速端到端视觉避障任务，目标是在密集障碍环境中实现安全、实时、可复现的闭环控制系统，并重点解决以下三个核心问题：

1. 如何设计高效的空间表征与时序聚合结构，以提升高速段避障鲁棒性与分布外泛化能力；
2. 如何保证序列模型在流式部署中的状态一致性，避免因错误状态管理导致无记忆退化与系统性漂移；
3. 如何在保持安全性的同时控制指令抖动代价，构建部署可用的平滑/约束机制。

1.4.2 技术路线概述

本文的技术路线由三个递进阶段组成，每个阶段对应一至两项核心研究内容。图 1.1 给出了技术路线总览。

各阶段的具体内容如下：

阶段 A：端到端系统设计——网络架构、训练方法与部署约束。本文采用端到端视觉控制框架：每个控制周期策略接收单目深度观测与轻量状态输入，输出世界坐标系下的速度指令，由仿真器/低层控制器执行形成闭环。为支撑大规模数据生成与可控评测，本文使用高保真仿真平台 Flightmare 进行训练与测试 [22]。在策略网络方面，

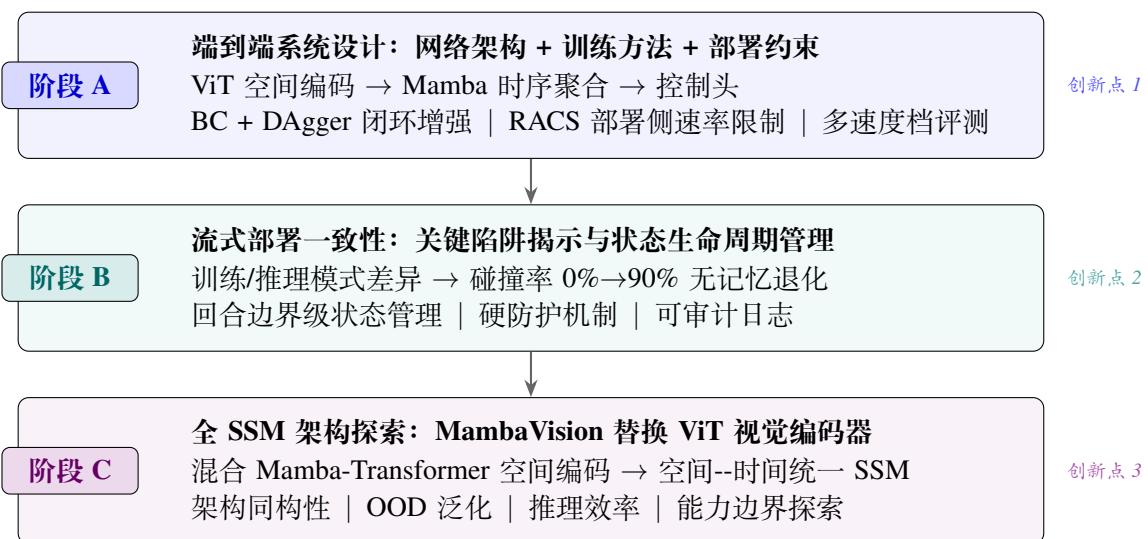


图 1.1 本文技术路线总览

以“空间编码 + 时序聚合 + 控制头”为基本架构：空间编码器采用 ViT[20] 提取空间表征，时序模块采用选择性状态空间模型 Mamba[15] 聚合时序信息，实现从单目深度与轻量状态到世界坐标速度指令的端到端映射。训练方面，首先采用行为克隆（BC）范式建立强基线；在此基础上引入 DAgger[21] 闭环数据增强（3 轮迭代），逐步缩小训练分布与部署分布之间的差距，降低碰撞频次并提升跨试验稳定性。为缓解敏捷避障带来的指令抖动代价，本文进一步设计部署侧动态速率限制控制平滑器（RACS），以最小工程复杂度换取显著的平滑性改善。DAgger 方法见第 4 章 4.8 节，RACS 方法见第 4 章 4.9 节，实验结果详见第 6 章。

阶段 B：流式部署一致性——关键陷阱揭示与状态生命周期管理。序列模型在流式部署中存在一个关键陷阱（Critical Pitfall）：训练与推理的模式差异可能导致内部状态在错误时刻被重置，使模型退化为“无记忆策略”。本文系统分析了该现象的成因与后果——实验表明，错误的逐步重置会使碰撞率从 0% 飙升至 90%——并提出回合边界级状态生命周期管理协议与硬防护机制（运行时断言、配置锁定与可审计日志），确保部署一致性与评测可信度。该发现对所有使用序列模型进行端到端控制的研究具有普遍警示意义。

阶段 C：全 SSM 架构探索——MambaVision 替换 ViT 视觉 backbone。在前两阶段确立的 ViT+Mamba 系统基础上，本文进一步探索将空间编码器从 ViT 替换为同属 SSM 系列的 MambaVision[16]，形成空间--时间统一的全 SSM 架构。该探索的核心价值不仅在于性能比较，更在于考察 SSM 在视觉感知领域的能力边界与空间--时间同

构建模的可行性。即使性能提升有限，该实验仍为理解 SSM 在端到端控制中的适用范围提供有价值的实证基础。

1.5 本文主要贡献与创新点

结合上述研究目标与技术路线，本文形成如下三项主要贡献与创新点：

1. 提出面向高速端到端避障的 ViT+Mamba 时序策略网络，构建 BC+DAgger+RACS 的完整训练--部署系统，并建立多速度档系统评测体系。方法：构建以 ViT 空间编码、Mamba 选择性状态空间模型时序聚合与线性控制头为核心的端到端策略网络。训练方面采用行为克隆（BC）建立强基线，并引入 DAgger 闭环数据增强缓解分布偏移；部署方面设计 RACS 动态速率限制模块控制指令抖动代价。验证：在 5 个速度档（3 m/s–12 m/s）与同分布（Spheres）/分布外（Trees）双环境下进行零样本评测。DAgger 实验验证碰撞频次与方差随迭代收敛；RACS 实验验证 Jerk 显著降低而安全性基本保持。（对应第 4、6 章）
2. 揭示序列模型端到端控制落地中的一个关键陷阱（Critical Pitfall）：流式部署状态管理错误导致碰撞率从 0% 飙升至 90%；提出回合边界级状态生命周期管理协议与硬防护机制。方法：系统分析训练模式（定长序列 batch 前向）与推理模式（逐步递推）的差异导致的状态错误重置问题；设计回合边界级状态生命周期管理协议——内部状态仅在回合开始时初始化、回合内保持连续传播；引入运行时断言、配置锁定与可审计日志作为硬防护机制。验证：通过 KeepState 与 ResetState 的对比实验，碰撞率从 0% 跳升至 90%、Mean Y Drift 从 0.022 m 增至 0.770 m，定量证实状态管理错误的毁灭性后果。该发现对所有使用序列模型进行端到端控制的研究具有普遍警示意义。（对应第 5 章）
3. 从混合架构走向全 SSM 架构的探索：将空间编码器从 ViT 替换为 MambaVision，量化空间--时间同构建模的可行性与能力边界。方法：在保持时序 Mamba 模块、训练流程与部署一致性机制完全不变的条件下，将视觉编码器替换为 MambaVision[16]（混合 Mamba-Transformer backbone），形成空间--时间统一的 SSM 系列架构。验证：在相同的多速度档与 OOD 场景下，对比 ViT 与 MambaVision 在碰撞率、OOD 泛化鲁棒性、推理延迟与显存占用四个维度的表现。核心价值：该探索的贡献在于提出并验证全 SSM 架构在端到端控制中的可行性，为

理解 SSM 在视觉--运动控制任务中的能力边界提供实证基础。即使性能提升有限，空间--时间同构性带来的架构简洁性与工程统一性仍具理论意义。（对应第 6 章控制变量实验）

1.6 论文结构安排

本文共分七章，各章内容安排如下：

第 1 章 绪论。介绍高速端到端视觉避障的研究背景与问题提出，阐述研究意义与应用价值，分析关键挑战（包括闭环分布偏移问题），给出研究内容与技术路线，总结本文主要贡献与创新点，并说明论文结构安排。

第 2 章 相关工作与研究现状。系统综述模块化自主飞行（感知--规划--控制范式）、端到端视觉飞行控制（从模仿学习到强化学习）、视觉表征与网络结构（CNN、ViT 与 MambaVision）、时序建模（LSTM、Transformer 与结构化状态空间模型）、以及安全性与部署侧约束机制等方面国内外研究进展，明确本文的切入点与定位。

第 3 章 问题定义与系统框架。给出高速端到端视觉避障任务的形式化定义，包括观测空间、动作空间、奖励/损失设计与评价指标；描述基于 Flightmare 仿真平台的系统架构、数据生成流程与闭环评测协议。

第 4 章 ViT+Mamba 策略网络与训练方法。详细介绍端到端策略网络的架构设计（ViT 空间编码器、Mamba 时序聚合模块、控制头）与基于行为克隆（BC）的训练流程，给出 DAgger 闭环数据增强的方法与实现细节，以及部署侧动态速率限制控制平滑器（RACS）的算法定义、数学形式与安全学习方法谱系定位。

第 5 章 流式部署一致性与状态生命周期管理。系统分析序列模型在流式推理中的状态一致性问题，揭示无记忆退化的关键陷阱（碰撞率从 0% 飙升至 90%），提出回合边界级状态管理协议与硬防护机制，并通过对比实验验证该机制对评测可信度的决定性影响。

第 6 章 实验设置与结果分析。给出完整的实验设置（环境配置、评测协议、基线对比与消融实验），在多速度档与多障碍分布下评估策略性能。在 BC 基线对比之后，依次给出 RACS 部署侧约束实验、DAgger 闭环数据增强实验的结果与分析，以及从混合架构走向全 SSM 架构的 MambaVision 探索实验框架设计。

第 7 章 总结与展望。总结全文研究内容与主要结论，讨论现有方法的局限性，并展望未来在真实环境部署、动态障碍应对、多模态融合等方面的拓展方向。

第 2 章 相关工作与国内外研究现状

2.1 模块化自主飞行：感知--规划--控制范式

模块化导航系统通常由状态估计/建图 (Perception)、路径/轨迹规划 (Planning) 与控制器 (Control) 三大模块串联构成。其优势在于可解释性强、模块边界清晰、各子系统可独立验证与替换；在结构化环境与中低速条件下，配合成熟的视觉/视觉惯性里程计 (VIO)、SLAM 与轨迹优化器可以取得可靠表现。然而，正如第 1 章所述，该范式在高速密集障碍环境中的串联延迟与跨模块误差传播会显著放大系统脆弱性。Loquercio 等在野外高速飞行工作中对这一问题给出了非常明确的动机与实验论证 [1]。本节分别从状态估计、路径规划与仿真平台三个层面综述模块化范式的典型方法与局限。

2.1.1 视觉/视觉惯性里程计与 SLAM

状态估计是模块化导航系统的基石，其精度与实时性直接决定后续规划与控制模块的可靠性。视觉 SLAM 与视觉惯性里程计 (VIO) 在无人机导航中被广泛采用，根据前端特征提取方式的不同，可分为基于特征点的稀疏方法与基于直接法的半稠密/稠密方法两大类。

在稀疏特征方法中，ORB-SLAM2 是最具代表性的开源 SLAM 系统之一 [2]。该系统支持单目、双目与 RGB-D 三种模态输入，通过 ORB 特征的快速提取与匹配实现鲁棒追踪，并集成回环检测与全局位姿图优化以消除累积漂移。ORB-SLAM2 在学术研究与工程原型中被广泛使用，其模块化设计（追踪线程、局部建图线程、回环线程）也成为后续 SLAM 系统的参考架构。然而，ORB-SLAM2 依赖纹理特征进行匹配，在弱纹理、重复纹理或快速运动导致的运动模糊条件下，特征提取与匹配质量会显著下降，从而影响定位精度与系统鲁棒性。

在视觉惯性估计方面，VINS-Mono 是单目视觉惯性紧耦合估计的代表性工作 [3]。该系统将视觉特征观测与 IMU 预积分在非线性优化框架中进行联合估计，并设计了鲁棒的在线初始化、故障检测与恢复机制，使其在资源受限的机载平台上也能实现高精度的实时状态估计。VINS-Mono 的紧耦合架构能够有效利用 IMU 的高频率与互补性来弥补视觉在快速运动时的信息缺失，这使得 VIO 成为中速自主飞行中较为可靠的状态估计方案。

从视觉 SLAM 面临的核心挑战来看，主要包括：（1）**尺度与初始化**——单目 SLAM 存在尺度模糊，需要通过运动先验或 IMU 融合来恢复度量尺度；（2）**动态物体**——动态场景中运动对象的特征会污染位姿估计，需要语义或几何手段进行筛选；（3）**回环检测**——回环检测的鲁棒性与计算效率直接影响长时间运行的漂移修正能力；（4）**运动模糊与快速旋转**——在高速飞行中尤为突出，图像模糊与大幅帧间运动会导致特征追踪失败。高翔等在“视觉 SLAM 研究进展”中从稀疏/半稠密/稠密地图构建、前端优化架构与开放问题等维度对国内外视觉 SLAM 研究进行了系统梳理 [23]。张弓等在“移动机器人视觉惯性 SLAM 研究进展”中则从滤波与优化两大路线总结了 VIO/VI-SLAM 的发展脉络与核心挑战 [24]。

总体而言，视觉 SLAM/VIO 在结构化环境与中低速条件下具有稳定优势；但在高速密集障碍场景中，运动模糊会严重削弱视觉前端质量，串联流水线的推理延迟会等效为状态预测偏差，而建图不完备则使规划器缺少足够信息以做出安全决策。这些因素叠加后，模块化系统的闭环鲁棒性会显著降低 [1]。

2.1.2 路径规划与轨迹规划

规划模块负责在感知模块构建的地图或障碍表示上生成安全可行的飞行路径或动力学可执行的轨迹。根据问题类型，可分为全局路径规划与局部轨迹优化/避障两个层面。

2.1.2.1 全局路径规划：采样方法与渐近最优

在全局路径规划中，采样方法因适用性强、可处理高维与复杂约束空间而被广泛采用。Karaman 与 Frazzoli 提出的 RRT* 与 PRM* 给出了渐近最优采样规划的理论基础 [4]，证明在采样数趋于无穷时路径代价收敛至全局最优，为后续大量采样规划工作奠定了理论框架。基于 RRT* 的改进工作在学术与工程中数量庞大，围绕采样策略、树结构优化、启发式引导与并行化等方面持续发展。

国内在三维路径规划方面同样积累了大量研究。王猛等针对无人机三维路径规划与避障问题，结合改进粒子群优化与滚动策略以实现实时避障与平滑性提升 [25]，反映了国内学界在智能优化方法与工程可用性方面的典型思路。李志刚等在多旋翼无人机避障航迹规划研究中，围绕 RRT 类算法改进以满足动力学与代价约束展开工作 [26]，体现了国内在采样规划工程化方面的持续投入。

2.1.2.2 局部避障与安全回退机制

在局部避障与轨迹优化层面，规划器往往需要在“速度”与“安全”之间进行折中。面向未知或部分已知环境中的高速飞行，安全回退是关键机制之一。FASTER 规划器强调在未知环境中同时维护快速探索轨迹与安全回退轨迹 (safe backup trajectory)，使得在前方未知区域被感知到障碍时，系统能够切换到已验证安全的回退轨迹，从而在提高速度上限的同时降低碰撞风险 [5]。这种“安全回退 + 轨迹优化”的设计思想在后续多项工作中得到继承与拓展。

总体上，规划类方法具有明确的可解释性与安全回退机制，在低速至中速结构化场景中表现可靠。但在高速极限工况下，规划器仍可能受限于三个因素：(1) 感知延迟导致地图更新滞后；(2) 建图不完备使得规划空间信息不足；(3) 在线重规划的计算预算受限。这些因素共同构成了端到端方法在高速场景中的替代动机。

2.1.3 高保真仿真平台与可复现实验基座

端到端学习控制高度依赖可规模化数据采集与可控实验条件，因此高保真仿真平台是关键基础设施。

Flightmare 是面向四旋翼研究的代表性仿真器之一 [22]。其设计强调渲染引擎与物理引擎的解耦：物理仿真可以在不启动渲染的情况下以极高速率运行（用于大规模数据生成与策略训练），也可以启动高保真渲染以支持视觉观测的生成与可视化评测。这种灵活性使 Flightmare 成为端到端视觉避障研究中广泛使用的实验平台。

另一方面，Foehn 等提出的 Agilicious 提供了开放软硬件一体化平台 [27]，覆盖从模型预测控制到神经网络控制的系统化验证需求。Agilicious 不仅开源了仿真与控制代码，还提供了配套硬件方案，使得敏捷飞行研究具备可复现性与工程闭环能力，是近年敏捷无人机研究的重要基础设施之一。

仿真平台的可用性对于端到端方法尤为重要：训练数据的规模与多样性、评测场景的可控性与可重复性、以及仿真到真实 (Sim-to-Real) 迁移的可行性，都直接影响研究结论的可信度。本文选择 Flightmare 作为主要实验平台，利用其高效物理仿真与可配置障碍场景支撑大规模训练与系统评测。

2.2 端到端视觉飞行控制：从轻量网络到高速野外与冠军竞速

端到端控制通常以" 视觉到控制 (vision-to-control) " 为核心，直接回归控制量或短期轨迹，避免复杂建图与规划。其关键难点在于如何在噪声观测与分布偏移下保持鲁棒，以及如何在闭环中避免累积误差导致发散。

2.2.1 模仿学习、特权信息专家与 Sim-to-Real

端到端视觉控制的早期探索可追溯到 DroNet[8]，该工作采用相对轻量的卷积神经网络从单目图像直接输出转向角与碰撞概率，实现了城市环境中的端到端导航，体现了" 网络简化 + 实时可用" 的早期思路。Deep Drone Racing[11] 则通过域随机化在仿真中训练并实现零样本迁移到真实四旋翼竞速环境，展示了" 仿真大数据 + 迁移鲁棒性" 的重要路线。CAD2RL[9] 进一步验证了纯合成数据训练后直接迁移到真实室内场景的可能性，为大规模仿真训练提供了早期证据。

更进一步，Loquercio 等在 Learning High-Speed Flight in the Wild 中将端到端方法推进到复杂自然与人造环境中的高速飞行 [1]。该工作的核心设计包括：(1) 使用可获取完整环境信息的" 特权信息专家" (Privileged Expert) 生成高质量示范轨迹；(2) 通过系统化的传感噪声建模与域随机化实现零样本仿真到现实迁移；(3) 端到端映射从深度图像直接输出速度指令，绕过显式建图与规划。该工作在森林、雪地、坍塌建筑等多种复杂环境中展示了高速飞行能力，系统阐明了高速场景下模块化流水线的局限，是端到端高速避障研究的重要里程碑。

与" 避障" 任务不同，Deep Drone Acrobatics[28] 面向的是极限机动动作（如翻滚、循环翻转等），同样使用仿真训练与特权信息示范来获得可迁移策略。该工作体现了端到端方法在高动态控制任务中的潜力与方法论共性：特权信息专家 + 仿真训练 + 域随机化的组合已成为端到端无人机控制的标准范式之一。

2.2.2 强化学习与竞速：从" 可飞" 到" 超越人类冠军"

随着深度强化学习的发展，端到端系统在竞速场景中不断接近并超越人类水平。Kaufmann 等提出的 Swift 系统 [12] 结合仿真深度强化学习与真实数据校正，在真实无人机竞速对抗中达到了与人类冠军同级甚至胜出的水平。Swift 系统的成功依赖于多个关键要素：(1) 大规模仿真中的分布式强化学习训练提供了充足的探索与策略

优化基础；（2）真实飞行数据用于校正仿真与现实之间的系统性偏差（Sim-to-Real Gap）；（3）系统化的工程实现确保了从仿真策略到真实飞行器的可靠部署。

Swift 的成功表明，在充分的仿真基础设施与工程支撑下，端到端系统不仅可以在简单场景中替代传统流水线，更能在极端动态条件下展现出超越人类操控的性能上限。这一成果也反映出“仿真学习 + 真实数据校正”已成为端到端控制从实验室走向真实部署的主流范式。

2.2.3 模仿学习方法谱系与闭环分布偏移问题

在端到端控制的训练方法中，行为克隆（Behavioral Cloning, BC）因训练稳定、样本效率高而被广泛采用。BC 的基本思路是将专家示范轨迹作为监督信号，通过回归损失直接训练策略网络。然而，BC 的典型问题是闭环分布偏移（covariate shift）：训练数据来自专家轨迹的状态分布，而策略部署后诱导的实际状态分布会偏离训练分布，小误差会随时间累积并可能导致系统发散。

Ross 等提出的 DAgger（Dataset Aggregation）[21] 通过迭代式数据集聚合来缓解这一问题：在每轮迭代中，使用当前策略在线采集数据并由专家标注，然后将新数据合并到训练集中重新训练策略。DAgger 的理论分析表明，通过 T 轮迭代后策略的期望损失上界与训练集上的损失呈线性关系，而非像 BC 那样随时间步二次增长。DAgger 是序列决策模仿学习中最经典的方法之一，后续的许多改进（如 SafeDAgger、EnsembleDAgger 等）都建立在其框架之上。

在高速避障任务中，利用特权信息专家生成高质量示范数据并配合 DAgger 类闭环数据增强策略是提升可靠性的常见做法。本文采用 BC 与特权专家的组合进行训练，并在分布外（OOD）环境上进行零样本测试。为避免通过目标域再训练引入评测偏倚，本文严格采用零样本测试协议：策略仅在训练分布内数据上学习，不以任何形式接触测试环境，从而确保泛化能力的评测结论具备可信性。本文的核心控制变量是“时序建模 + 部署一致性”，而非数据泄漏或训练技巧。

2.3 时序建模：RNN/LSTM、Transformer 与结构化状态空间模型

高速避障不是静态映射问题：无人机控制回路必须利用短期历史来抑制观测噪声、预测障碍相对运动趋势并稳定闭环输出，因此时序建模是端到端策略的核心能力

之一。本节从 RNN/LSTM、Transformer 到结构化状态空间模型（SSM）的演进线索综述时序建模方法。

2.3.1 RNN/LSTM 的流式优势与局限

循环神经网络（RNN）及其变体长短期记忆网络（LSTM）[13] 是最早被用于端到端控制中时序信息聚合的模型。LSTM 通过门控机制（输入门、遗忘门、输出门）选择性地保留与更新记忆状态，缓解了普通 RNN 在长序列训练中的梯度衰减问题，使其能够在一定范围内维持对历史信息的有效记忆。

在在线控制场景中，LSTM/RNN 的天然优势在于支持流式递推推理：每个时间步只需输入当前观测并更新固定大小的隐状态，计算开销恒定且不随序列长度增长。然而，LSTM 也面临明确的局限：（1）**长期依赖建模受限**——尽管门控机制缓解了梯度衰减，但对于真正的长程依赖关系（如数百步以上），LSTM 的记忆能力仍然有限；（2）**训练效率**——RNN 的序列依赖性使其无法像 Transformer 那样进行完全并行化训练，在长序列上训练速度较慢；（3）**部署状态管理敏感性**——这是一个在实践中往往被忽视但影响深远的问题。

关于状态管理敏感性，需要特别指出：LSTM 在在线部署时高度依赖内部隐状态 $(\mathbf{h}_t, \mathbf{c}_t)$ 的正确传播。训练通常采用固定长度序列的 batch 前向传播，而在线推理则是逐步更新。如果工程实现中在错误时刻（如每次推理调用时）重置隐状态，模型将退化为“无记忆策略”——即每个时间步仅根据当前单帧观测做出决策，完全丧失了时序聚合能力。这种退化往往不会在离线指标上直接暴露，但会在真实闭环中导致系统性漂移与性能崩坏。本文将在第 5 章中系统分析这一问题并给出工程解决方案。

2.3.2 Transformer 与视觉 Transformer（ViT）

Transformer[14] 通过自注意力机制（Self-Attention）建模序列中任意位置对之间的依赖关系，彻底摆脱了 RNN 的序列递推约束，实现了完全并行化训练。其核心优势在于：（1）对长程依赖的建模能力更强——注意力权重直接连接任意两个位置，无需通过中间状态传递；（2）训练高度并行化——所有时间步可同时计算注意力与前馈输出。然而，自注意力机制的计算复杂度为 $O(n^2)$ (n 为序列长度)，在长序列或高分辨率输入下可能成为计算瓶颈。

在视觉领域，Dosovitskiy 等提出的 Vision Transformer（ViT）[20] 将 Transformer

范式引入图像识别：将图像划分为固定大小的 patch token，经线性映射后输入标准 Transformer 编码器。结合大规模数据预训练，ViT 在多项视觉基准上取得了优异表现，展示了 Transformer 在视觉表征方面的强大能力。

在四旋翼端到端避障方向，Xing 等 [29] 系统比较了 CNN、U-Net、循环结构与 ViT 等多种视觉 backbone 对避障性能的影响，指出在高速与泛化条件下 ViT 具备明显优势，且在 ViT 后加入循环模块（recurrence）能进一步改善时序表现。该工作的输出语义采用世界坐标系下的速度指令，与本文的系统设定具有较高一致性，为本文选择 ViT 作为空间编码器提供了直接的实验依据。

2.3.3 结构化状态空间模型：S4 与 Mamba

结构化状态空间模型（Structured State Space Model, SSM）是近年来序列建模领域的重要方向，其核心思想是基于连续时间线性状态空间方程对序列数据进行建模：

$$\mathbf{h}'(t) = \mathbf{Ah}(t) + \mathbf{Bx}(t), \quad \mathbf{y}(t) = \mathbf{Ch}(t) + \mathbf{Dx}(t) \quad (2.1)$$

其中 $\mathbf{h}(t)$ 为隐状态， $\mathbf{x}(t)$ 为输入， $\mathbf{y}(t)$ 为输出， $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ 为系统参数矩阵。

S4 (Structured State Spaces for Sequence Modeling) [30] 通过对矩阵 \mathbf{A} 的结构化参数化（如 HiPPO 初始化与对角化分解），使得 SSM 能够在长序列上高效训练，同时保持 $O(n)$ 的计算复杂度。S4 在长程依赖基准（如 Long Range Arena）上取得了突破性表现，成为 ICLR 2022 的代表性工作之一。

在 S4 的基础上，Gu 与 Dao 提出了 Mamba[15]，引入选择性机制（Selective Mechanism）：使 SSM 的参数 \mathbf{B}, \mathbf{C} 与离散化步长 Δ 依赖于输入内容，从而增强模型对不同输入的自适应选择能力。此外，Mamba 设计了硬件友好的并行扫描算法，在保持线性复杂度 $O(n)$ 的同时实现了高吞吐推理。与 Transformer 相比，Mamba 在长序列上的推理速度可达数倍提升，且显存占用更低。

对于高速闭环控制场景，Mamba 具有双重价值。在算法层面，选择性 SSM 能够根据输入内容动态调整状态更新幅度，对短期历史中的关键信息（如障碍距离突变、急转弯前的预兆）进行自适应聚合。在工程层面，Mamba 天生支持流式递推推理——离散化后的状态更新方程 $\mathbf{h}_t = \bar{\mathbf{A}}\mathbf{h}_{t-1} + \bar{\mathbf{B}}\mathbf{x}_t$ 与 LSTM 具有相似的递推形式，每个控制周期仅需一次矩阵运算即可完成状态更新。然而，这也意味着 Mamba 在流式部署时

同样依赖内部状态的正确持续传播，其状态生命周期管理必须严格与回合边界对齐。围绕这一问题建立可审计、可复现的评测范式，是将 SSM 用于机器人端到端控制的重要工程前提。

2.4 视觉 Backbone 演进：CNN→ViT→MambaVision

端到端避障策略的前端视觉编码器决定了对障碍结构、纹理变化、遮挡与噪声的表征上限。视觉 backbone 的演进经历了从 CNN 到 ViT 再到混合架构的过程。

2.4.1 卷积神经网络（CNN）

卷积神经网络 [31] 通过局部感受野、权重共享与层级特征提取建立了视觉表征的基础范式。ResNet[32] 引入的残差连接使得训练更深的网络成为可能，在图像分类、目标检测与语义分割等任务上取得了广泛成功。CNN 的优势在于：(1) 局部归纳偏置使其在小数据量时也具备较好泛化；(2) 计算高度并行且部署效率高；(3) 工程成熟度高，工具链完善。在早期端到端避障工作中，CNN 是默认的视觉编码器选择 [8, 9]。

然而，CNN 在全局结构关系建模方面受限于感受野大小：即便通过深层堆叠扩大感受野，信息仍需逐层传递，对于需要全局上下文（如远处障碍与当前运动方向的关系）的场景可能存在不足。

2.4.2 视觉 Transformer（ViT）的表征优势

ViT[20] 通过将图像切分为 patch token 并用 Transformer 编码器处理，以自注意力机制建模任意 patch 对之间的全局依赖，突破了 CNN 的感受野限制。在大规模预训练数据支持下，ViT 在多项视觉基准上达到或超越了 CNN 的表现。

对于端到端避障任务，ViT 的全局注意力机制在以下场景中具有潜在优势：(1) 复杂障碍布局需要全局结构理解（如多个障碍的相对位置关系）；(2) 纹理变化、光照变化与遮挡条件下需要更鲁棒的表征；(3) 分布外场景中需要更强的泛化能力。Xing 等 [29] 的实验比较也支持了 ViT 在高速避障任务中的优势。

然而，ViT 的 $O(n^2)$ 自注意力复杂度 (n 为 patch 数量) 在高分辨率输入下可能成为推理瓶颈，尤其是在机载算力受限的部署场景中。这构成了进一步探索更高效视觉 backbone 的动机。

2.4.3 MambaVision：混合 Mamba-Transformer 视觉骨干

近期，Hatamizadeh 与 Kautz 提出 MambaVision[16]：一种针对视觉应用定制的混合 Mamba-Transformer backbone。MambaVision 的核心设计包括：(1) 对 Mamba 模块进行面向视觉特征建模的重新设计，使其更适合二维空间结构的处理；(2) 通过系统化消融实验证在不同阶段融合 ViT 自注意力块与 Mamba 块的最优配比；(3) 在 ImageNet 分类、COCO 检测与 ADE20K 分割等多项基准上展示了优于纯 ViT 与纯 Mamba 方案的效率-精度权衡。MambaVision 于 CVPR 2025 发表，其代码与预训练模型已开源，为工程复现与任务迁移提供了直接入口。

对于本文的高速端到端避障系统而言，将空间编码器替换为 MambaVision 的动机包括三个方面。首先，在**效率**方面，MambaVision 在同等精度下相较纯 ViT 可降低计算量与推理延迟，这对机载部署的时间预算至关重要。其次，在**架构统一性**方面，本文已采用 Mamba 作为时序聚合模块；若空间编码器也采用 MambaVision（其中包含 Mamba 组件），则整体架构将形成更统一的“SSM 友好”结构，有助于减少架构异质性与工程复杂度。最后，在**泛化能力**方面，MambaVision 的混合注意力 +SSM 设计是否能在分布外场景中提供更鲁棒的表征，是本文需要通过控制变量实验证的关键假设。上述收益均需通过严格实验评测确认，而非先验假设。

2.5 安全性、平滑性与部署侧约束机制

学习型控制在真实部署中的核心顾虑之一是安全性：策略的输出可能因训练数据分布的局限、模型泛化的不足或环境扰动而产生不安全动作。同时，在高速避障中，更敏捷的策略可能伴随更高频率的控制指令抖动 (jerk)，影响执行器寿命、能耗与飞行平滑性。

2.5.1 安全学习控制研究概述

Brunke 等在 Annual Review 的综述中系统总结了从学习控制到安全强化学习的主要路线、挑战与开放问题 [17]。总体而言，安全方法可分为三大类别：(1) **训练时约束**——在目标函数或策略更新中引入安全惩罚、约束优化或拉格朗日对偶方法，使策略在训练过程中就倾向于满足安全约束；(2) **运行时证书与滤波**——对策略输出进行可行性检查与最小修改，确保实际执行的动作满足安全约束；(3) **基于模型的安全回**

退与混合控制——在学习策略不可信时切换到经验证安全的备份控制器。每种路线各有优劣：训练时约束可能限制策略探索空间，运行时滤波需要可靠的安全集估计，而混合控制则需要安全控制器的设计与切换逻辑。

2.5.2 控制障碍函数与 MPC 证书

控制障碍函数 (Control Barrier Function, CBF) 为系统安全约束提供了可证明的形式化工具。CBF 定义了一个安全集 $\mathcal{C} = \{\mathbf{x} : h(\mathbf{x}) \geq 0\}$ ，并通过约束 $\dot{h}(\mathbf{x}, \mathbf{u}) + \alpha(h(\mathbf{x})) \geq 0$ 保证系统状态始终保持在安全集内。Cheng 等提出 RL-CBF 框架 [18]，将 CBF 安全约束嵌入强化学习的策略优化过程中，使得学习到的策略在追求任务目标的同时满足安全性。

另一方面，Wabersich 与 Zeilinger 提出 MPSC (Model Predictive Safety Certification) 框架 [19]，采用不同的思路：不修改策略训练过程，而是在部署时对策略输出进行 MPC 可行性证书验证——若策略的候选动作满足约束则直接执行，否则求解一个最小修改优化问题以找到既满足安全约束又尽量接近原始策略输出的动作。MPSC 的优势在于对策略训练无侵入性，可作为“即插即用”的安全层。

2.5.3 部署侧平滑与速率限制

在高速端到端避障的实际部署中，策略的敏捷性提升可能伴随更高的指令变化率。过大的 jerk 不仅影响执行器寿命与能耗，还会导致飞行器姿态振荡，降低可控性。与训练侧约束相比，部署侧平滑机制具有实现简单、可调可控、对策略训练无侵入等工程优势。

常见的部署侧策略包括：（1）**速率限制**——限制相邻两个控制周期之间控制量变化的最大幅度；（2）**低通滤波**——对策略输出进行时域平滑，抑制高频抖动；（3）**风险自适应约束**——根据当前环境风险等级动态调整约束强度（高风险区放宽限制以保留敏捷性，低风险区增强平滑以节省能耗）。本文的 RACS (Rate-Adaptive Control Smoother) 模块属于第一与第三种路线的结合：当前版本以动态速率限制为核心，根据最小深度观测值调整相邻控制周期间的指令变化上界约束——障碍接近时放宽限制以保留敏捷性，远离障碍时增强平滑以降低抖动。该模块定位为部署侧的轻量平滑增强组件，虽不提供严格可证明的安全性保证，但通过可控的指令变化约束在实际系统中有效改善 jerk 并增强可用性，后续可扩展为基于碰撞概率或 TTC 的风险自适应

版本。

2.6 国内外研究现状总结与本文切入点

在展开国内外研究现状讨论之前，表 2.1 从系统范式、训练方法、时序建模与视觉编码四个维度对本章涉及的主要技术路线进行了横向对比，以帮助读者快速把握各路线的核心特征与适用场景。

表 2.1 高速端到端视觉避障相关技术路线对比

对比维度	路线 A	路线 B	A 的优势	B 的优势
系统范式	模块化（感知--规划--控制）	端到端（视觉 → 控制）	可解释、可验证、模块可替换	低延迟、架构简洁、可吸收大规模仿真数据
训练方法	行为克隆（BC）	强化学习（RL）	训练稳定、样本高效、实现简单	可在线探索、策略上限更高、可优化长期回报
时序建模	LSTM/RNN	SSM（Mamba）	流式推理天然支持、工程成熟	线性复杂度、高吞吐、选择性机制增强内容感知
视觉编码	ViT	MambaVision	全局注意力、强表征、大规模预训练生态	效率--精度更优、与时序 SSM 架构统一、部署友好

2.6.1 国外研究趋势

国际上，敏捷无人机研究在近年呈现出“系统化验证 + 端到端方法 + 高性能仿真平台支撑”的清晰趋势：

- 端到端高速飞行与野外复杂场景验证推动了方法从实验室走向真实落地 [1]。特权信息专家 + 仿真训练 + 域随机化的组合已成为标准范式。
- 强化学习在竞速对抗中达到人类冠军级别 [12]，强调“仿真学习 + 真实数据校正”的工程闭环。
- 开源平台（如 Agilicious[27]）与高性能仿真器（如 Flightmare[22]）持续降低研究门槛并提升可复现性。

- 在网络结构方面，视觉编码从 CNN 逐步走向 ViT[20]，并进一步出现 MambaVision[16] 等混合骨干；时序建模从 LSTM 走向结构化状态空间模型（S4[30]、Mamba[15]），“更高效的表征与建模”正在成为新的关注焦点。

2.6.2 国内研究现状

国内在无人机自主导航相关技术方面积累深厚，以下按三条典型路线进行归纳式综述。

2.6.2.1 三维路径规划与轨迹优化工程化

国内在采样规划与轨迹优化的工程化方面投入持续。在采样规划层面，李志刚等针对多旋翼无人机避障航迹规划提出改进 RRT 算法，引入动力学约束与代价优化以提升工程可用性 [26]；王猛等结合改进粒子群优化与滚动策略实现三维路径规划中的实时避障与平滑性提升 [25]。在轨迹优化层面，浙江大学高飞团队的系列工作具有代表性：Zhou 等提出 Fast-Planner，通过 B 样条参数化与梯度优化实现高效四旋翼轨迹生成 [33]；进一步的 EGO-Planner 摆脱了对欧氏有符号距离场（ESDF）的依赖，直接以障碍信息驱动梯度优化，在未知环境中实现了更高效的局部规划 [34]。何承坤等在四旋翼轨迹优化与跟踪控制综述中系统梳理了国内外在最优控制、微分平坦与数值优化三条技术路线上研究进展 [35]。总体而言，国内在规划工程化方面的研究活跃度高、成果丰富，但多数工作仍以模块化范式为前提，较少涉及端到端框架下的规划能力。

2.6.2.2 视觉 SLAM/VIO 工程部署

视觉 SLAM 与 VIO 是无人机自主导航的基础能力模块。高翔等在“视觉 SLAM 研究进展”中从稀疏/半稠密/稠密地图构建、前后端优化架构到开放问题进行了系统梳理 [23]；张弓等从滤波与优化两大技术路线总结了移动机器人视觉惯性 SLAM 的发展脉络 [24]。值得注意的是，VINS-Mono[3] 虽以英文发表，但由港科大秦通等中国学者主导，其紧耦合 VIO 方案已成为国内外无人机状态估计的标准参考实现之一。国内在 VIO/VI-SLAM 的系统综述、算法改进与嵌入式部署方面积累扎实，为模块化导航提供了可靠的状态估计基座；但这些方法在高速运动模糊与极限工况下仍面临本

章前述的固有局限。

2.6.2.3 学习型避障与端到端方法探索

国内在学习型避障方面的研究近年来增长迅速。雷志勇等基于深度强化学习探索了视觉避障方法 [36]，代表了国内将 DRL 应用于无人机避障的早期工作。陈杰等在“深度强化学习在无人机自主导航中的研究进展”综述中系统总结了 DRL 从离散动作空间到连续控制、从仿真训练到 Sim-to-Real 迁移的技术演进，并对国内相关课题组的代表性成果进行了归纳 [37]。总体来看，国内在学习型避障方面呈现以下特征：(1) 以深度强化学习路线为主，模仿学习方面的工作相对较少；(2) 多数工作以低速或中速场景为验证条件，对高速 ($> 7 \text{ m/s}$) 密集障碍环境的系统化闭环验证尚不充分；(3) 对流式部署一致性、状态生命周期管理等工程可信性问题的关注较少。这些空白为本文的研究切入提供了明确的定位依据。

2.6.3 研究趋势与本文切入点

综合国内外现状，可以归纳出三个清晰趋势：

1. **更高速、更复杂环境、更强闭环鲁棒性**成为评价标准——方法的有效性需要在高速密集障碍、分布外场景等挑战性条件下得到验证。
2. **端到端方法从“能跑”走向“可复现、可审计、可部署”**——尤其是流式推理一致性、状态生命周期管理与工程防护等问题受到越来越多关注。
3. **视觉 backbone 与时序模型同时演进**——ViT 推动了表征升级，SSM (S4/Mamba) 推动了高效序列建模，MambaVision 进一步把 SSM 思想带入视觉 backbone，为“空间--时间统一建模”提供了新方向。

结合上述趋势与高速避障的关键挑战，本文的切入点在于：围绕“时序建模能力 + 流式部署一致性 + 部署侧约束”构建端到端闭环系统，并在此基础上探索将空间编码器升级为 MambaVision 的可行性与收益边界。本文强调严格控制变量与可审计实现细节，以确保评测结论可信、可复现，并为后续工程部署与架构选型提供系统性依据。

第3章 问题定义与系统框架

本章对高速端到端视觉避障任务进行形式化定义，明确观测空间、动作空间、回合终止条件与评价指标，并描述基于 Flightmare 仿真平台的闭环控制架构、特权信息专家数据生成流程以及可审计的评测协议。本章所建立的定义与协议将贯穿后续所有实验章节，确保评测结论的可复现性与可信性。

3.1 任务定义与回合终止条件

3.1.1 任务形式化

本文研究的任务为四旋翼在三维密集障碍环境中的高速视觉避障。该任务可形式化为一个序列决策问题：在每个控制周期 t ，策略 π 根据当前观测 o_t 输出控制动作 a_t ，由仿真器或低层控制器执行后产生下一时刻的观测 o_{t+1} ，形成闭环。形式化地，该任务由以下五元组定义：

$$\mathcal{M} = \langle \mathcal{O}, \mathcal{A}, \mathcal{T}, \mathcal{G}, \tau_{\max} \rangle \quad (3.1)$$

其中 \mathcal{O} 为观测空间（包含视觉观测与轻量状态）， \mathcal{A} 为动作空间（世界坐标系下的速度指令）， $\mathcal{T}: \mathcal{O} \times \mathcal{A} \rightarrow \mathcal{O}$ 为由仿真器物理引擎决定的状态转移函数， \mathcal{G} 为回合终止条件集合， τ_{\max} 为最大回合时长。

3.1.2 评测环境

评测环境包含两类障碍分布，用于分别验证同分布性能与分布外泛化能力：

1. **Spheres**（同分布环境）：三维空间中随机分布的球体障碍，障碍物的位置、大小与密度在训练数据生成时已被覆盖。该环境作为策略的同分布测试条件。
2. **Trees**（分布外环境）：树状结构障碍，其几何形态（细长圆柱与冠层）与训练时的球体障碍存在显著差异。该环境用于检验策略在未见过的障碍形态下的零样本泛化能力。

3.1.3 回合终止条件

每个回合 (Trial) 的终止由以下条件共同确定：

- **到达终点**: 无人机沿 X 轴 (主飞行方向) 的累积飞行距离超过 58m 至 60m 时，判定到达终点线，回合正常结束。
- **超时终止**: 系统设置 $\tau_{\max} = 40\text{s}$ 的硬性时间上限。若在此时间内未到达终点，回合因超时而终止。

需要特别强调的是：**碰撞不会立即终止回合**。碰撞标志在整个回合持续记录，用于统计全程尺度的碰撞频率与碰撞事件次数。这一设计使得评测能够反映策略在碰撞后的恢复能力，而非仅度量“首次碰撞前飞行距离”。

3.2 观测空间与动作空间

3.2.1 深度图像观测

在每个控制周期 t ，策略接收单目深度图像 $D_t \in \mathbb{R}^{H \times W}$ 作为视觉输入。深度值以米为单位表示。图像分辨率设置为 $H = 60, W = 90$ ，并在输入策略网络前进行以下预处理：

1. 将原始深度值乘以缩放因子 $\alpha = 0.09$ 进行归一化，使数值范围适配网络训练；
2. 训练阶段引入高斯噪声 ($\sigma = 0.02$) 与随机亮度扰动 ($\pm 10\%$) 以增强策略对传感噪声的鲁棒性。

3.2.2 轻量状态输入

除视觉观测外，策略还接收轻量状态向量 s_t ：

$$s_t = [q_t, \tilde{v}^{\text{target}}] \quad (3.2)$$

其中：

- $q_t = [w, x, y, z]$ 为无人机在世界坐标系下的实时姿态单位四元数，采用 $[w, x, y, z]$ 排列顺序；

- $\tilde{v}^{\text{target}} = v^{\text{target}} / 10$ 为目标前向速度的归一化输入，通过线性缩放将速度值映射至与四元数量级相近的范围，有利于训练稳定性。

策略网络不直接输入无人机的实时速度，而是以目标速度作为条件输入。这一设计的考虑是：策略应学习根据视觉观测与姿态信息在障碍环境中维持目标速度并完成避障，而非依赖实时速度反馈进行简单的速度跟踪。目标速度作为条件输入允许同一策略在不同速度档位下评测，而不需要为每个速度单独训练模型。

3.2.3 动作空间

策略在每个控制周期输出世界坐标系下的三维线速度指令：

$$\mathbf{v}_t = [v_t^x, v_t^y, v_t^z] \in \mathbb{R}^3 \quad (3.3)$$

采用世界坐标系（world frame）输出的原因是：与对比基线保持相同的控制语义，确保 ViT+Mamba 与 ViT+LSTM 在公平条件下进行比较。该速度指令经由低层控制器转化为电机指令，由仿真器执行并更新无人机状态。

3.3 闭环控制回路与部署形态

3.3.1 系统架构

本文采用的端到端闭环控制系统由三个层次组成：感知层、策略层与执行层。图 3.1 给出了闭环控制回路的时序示意。

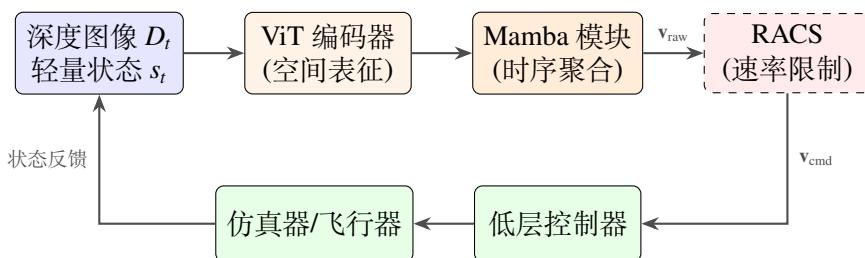


图 3.1 端到端闭环控制回路时序示意

在每个控制周期内，系统执行以下流程：

1. 仿真器/飞行器提供当前深度图像 D_t 与轻量状态 s_t ；

2. 视觉编码器（ViT）将深度图像编码为空间特征向量；
3. 时序聚合模块（Mamba）融合空间特征与轻量状态，结合内部时序状态输出原始速度指令 \mathbf{v}_{raw} ；
4. 部署侧约束模块（RACS，可选）对指令施加动态速率限制，输出最终指令 \mathbf{v}_{cmd} ；
5. 低层控制器将速度指令转化为电机指令并执行，更新无人机状态。

3.3.2 仿真平台

本文所有实验在 Flightmare 高保真仿真平台 [22] 中完成。Flightmare 的设计强调物理引擎与渲染引擎的解耦：物理仿真可以在不启动渲染的情况下以极高速率运行（用于大规模数据生成），也可以启动渲染以支持视觉观测生成与可视化评测。本文利用 Flightmare 的以下特性：

- 高效物理仿真支撑大规模专家数据生成；
- 可配置障碍场景（Spheres、Trees 等）支撑多分布评测；
- 精确的碰撞检测与状态记录支撑帧级指标统计。

3.3.3 控制频率与延迟预算

系统以策略网络的推理周期为基本控制频率运行。在本文的硬件配置（NVIDIA RTX 4060 GPU）下，ViT+Mamba 策略的单步推理时间为毫秒级，可满足高速飞行所需的控制带宽。控制周期的实际分布（包括推理时间与系统调度抖动）将在第 6 章中通过 Δt 分布统计进行分析，以排除系统负载差异对实验结论的混淆影响。

3.4 特权信息专家与数据生成

3.4.1 专家策略设计

本文采用行为克隆（Behavioral Cloning）范式训练策略网络，示范数据由带特权信息的专家策略生成。与学生策略仅能获取深度图像不同，专家策略在每个控制步可访问以下特权信息：

- 无人机的完整状态（位置、速度、姿态）；
- 一定局部范围内障碍物的精确几何信息。

专家策略的决策过程如算法 1 所示。

Algorithm 1 特权信息专家策略

Require: 无人机状态（位置 \mathbf{p} 、姿态 q ）、局部障碍几何、目标速度 v^{target} 、前视距离 d_{look}

Ensure: 世界坐标系下的速度指令 $\mathbf{v}_{\text{expert}}$

- 1: 在无人机前方 d_{look} 处的 $y-z$ 平面上均匀离散采样候选航点集合 $\mathcal{W} = \{w_1, w_2, \dots, w_K\}$
- 2: **for** 每个候选航点 $w_i \in \mathcal{W}$ **do**
- 3: 从当前位置 \mathbf{p} 到 w_i 执行直线碰撞检测
- 4: **if** 路径无碰撞 **then**
- 5: 标记 w_i 为可行航点
- 6: **end if**
- 7: **end for**
- 8: 从所有可行航点中选择最接近网格中心的航点 w^*
- 9: 计算相对位移 $\Delta\mathbf{p} = w^* - \mathbf{p}$
- 10: 施加比例增益生成速度指令 $\mathbf{v}_{\text{expert}} = K_p \cdot \Delta\mathbf{p}$
- 11: **return** $\mathbf{v}_{\text{expert}}$

3.4.2 训练数据集

训练数据集仅在 **Spheres** 环境中生成，包含约 585 条专家轨迹。学生策略以深度图像 D_t 与轻量状态 s_t 为输入，以专家速度指令 $\mathbf{v}_{\text{expert}}$ 为监督信号进行回归学习。

为验证策略的泛化能力，所有策略网络仅在 Spheres 环境生成的专家数据上训练，并在 Trees 环境中进行零样本（Zero-shot）测试——策略从未接触过 Trees 环境的任何数据。这一严格的评测协议确保了泛化能力评估的公正性：性能差异完全来源于策略的内在泛化能力，而非数据泄漏或目标域再训练。

3.5 评价指标与统计协议

3.5.1 安全性指标

(1) 全程碰撞率（Collision Rate）。定义为回合内碰撞帧数占回合总帧数的比例：

$$\text{Collision Rate} = \frac{\sum_{t=1}^T \mathbb{1}[\text{collision}_t = 1]}{T} \quad (3.4)$$

其中 T 为回合总帧数, $\text{collision}_t \in \{0, 1\}$ 为第 t 帧的碰撞标志。该指标度量碰撞接触在整个飞行过程中的频繁程度与持续时间。

(2) 碰撞事件次数 (Collision Count)。将连续碰撞帧视为同一次碰撞事件, 统计碰撞标志从 0 变为 1 的上升沿次数:

$$\text{Collision Count} = \sum_{t=2}^T \mathbb{1}[\text{collision}_t = 1 \wedge \text{collision}_{t-1} = 0] \quad (3.5)$$

该指标刻画独立碰撞事件的发生频次, 与 Collision Rate 互补。

(3) 成功率 (Success Rate)。定义为在超时限 τ_{\max} 内到达终点线的回合比例:

$$\text{Success Rate} = \frac{\text{到达终点的回合数}}{\text{总回合数}} \quad (3.6)$$

(4) 超时率 (Timeout Rate)。定义为因超时而终止的回合比例:

$$\text{Timeout Rate} = 1 - \text{Success Rate} \quad (3.7)$$

3.5.2 平滑性指标

指令抖动 (Command Jerk)。定义为相邻两个控制步发布的速度指令之差的 L_2 范数:

$$\text{Jerk}_t = \|\mathbf{v}_t - \mathbf{v}_{t-1}\|_2 \quad (3.8)$$

报告回合内平均值 $\overline{\text{Jerk}} = \frac{1}{T-1} \sum_{t=2}^T \text{Jerk}_t$ 及跨回合统计量。需要指出的是: 若启用 RACS 部署侧约束模块, 则以最终发布并执行的速度指令 \mathbf{v}_{cmd} (而非网络原始输出 \mathbf{v}_{raw}) 计算 jerk, 以反映真实控制平滑性。

3.5.3 系统性能指标

推理时间 (Inference Time)。记录单步模型前向推理耗时, 用于评估策略的实时性与部署可行性。

3.5.4 统计方式

对每个速度档位与环境配置下的 10 次独立试验，报告各指标的均值与标准差。不同方法之间的性能差异通过均值对比与方差分析进行评估。

3.6 评测可审计规范

为确保实验结论的可复现性与可追溯性，本文建立以下评测可审计规范：

1. **随机种子固定**：所有实验固定随机种子（包括 PyTorch、NumPy、CUDA 确定性模式与环境初始化种子），确保同一配置下的实验结果可精确复现。
2. **环境参数记录**：每次评测自动记录环境类型（Spheres/Trees）、障碍密度参数、目标速度档位与回合终止条件等关键配置。
3. **状态重置时机**：明确记录序列模型内部状态的重置时机（仅在回合边界），并通过运行时断言确保回合内状态的连续传播（详见第 5 章）。
4. **日志字段**：每次试验的日志包含请求配置与实际生效配置的对比记录，确保不存在配置被意外覆盖的情况。
5. **版本号固化**：记录策略网络权重文件的哈希值、代码版本号与依赖库版本，使得实验环境可完整还原。
6. **控制周期分布**：记录每次试验中所有控制步的 Δt 时间间隔分布，用于排除系统负载差异造成的混淆因素（详见第 6 章分析）。

上述规范贯穿本文所有实验，确保评测结论不受实现细节污染，并为后续研究者提供可复现的评测基线。

第4章 ViT+Mamba 策略网络与训练方法

本章详细介绍端到端策略网络的架构设计与训练方法。策略采用“空间编码 + 时序聚合 + 控制头”的三段式结构：视觉编码器（ViT）提取空间表征，时序模块（Mamba）融合历史信息，控制头输出速度指令。训练采用行为克隆（BC）范式，以三阶段课程学习策略平衡模仿精度与控制平滑性。本章同时给出对比基线的配置与 DAgger 数据增强的扩展方案。

4.1 总体架构

4.1.1 三层系统架构

本文的端到端策略网络如图 4.1 所示，由感知层、策略层和执行层三部分组成：

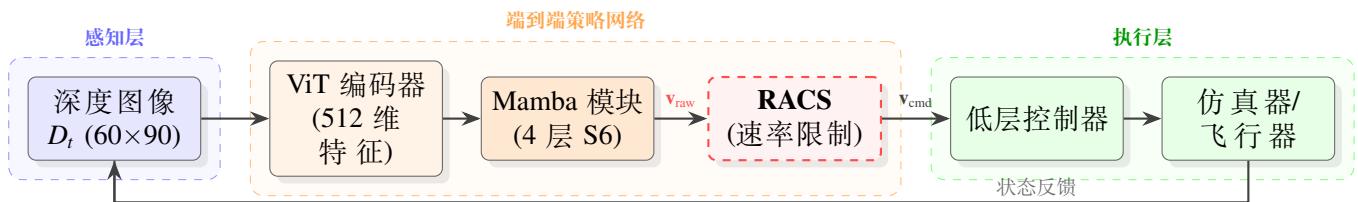


图 4.1 端到端导航管线架构总览

- **感知层**: 提供单目深度图像 D_t 与轻量状态 s_t ;
- **策略层**: ViT 编码器提取空间特征，Mamba 模块进行时序聚合，控制头输出速度指令；RACS 模块（可选）在部署侧施加速率限制；
- **执行层**: 低层控制器将速度指令转化为电机控制量，由仿真器或真实飞行器执行。

4.2 视觉编码器：轻量化 ViT

视觉编码器负责将深度图像 D_t 压缩为固定维度的空间特征向量。本文采用轻量化的 2-stage ViT 架构，在保持足够表征能力的同时满足高速飞行的实时性要求。

4.2.1 架构设计

2-stage ViT 的编码过程如下：

1. 第一阶段：使用 Patch Size 7×7 、Stride 4 的卷积嵌入层，将 60×90 的深度图像分割为 16×24 的 patch token 序列，通道数为 32；经过 Transformer 编码块处理后输出空间特征图；
2. 第二阶段：使用 Patch Size 3×3 、Stride 2 的卷积嵌入层进一步下采样至 8×12 ，通道数升至 64；经过 Transformer 编码块后将特征池化为 512 维向量。

表 4.1 给出了编码过程中各阶段的张量尺寸变化。

表 4.1 ViT 编码器各阶段张量尺寸

阶段	输入尺寸	Patch/Stride	输出尺寸	通道数
输入深度图	$60 \times 90 \times 1$	—	—	1
Stage 1 嵌入	60×90	$7 \times 7 / \text{Stride } 4$	16×24	32
Stage 1 编码	$16 \times 24 \times 32$	—	$16 \times 24 \times 32$	32
Stage 2 嵌入	$16 \times 24 \times 32$	$3 \times 3 / \text{Stride } 2$	$8 \times 12 \times 64$	64
Stage 2 编码	$8 \times 12 \times 64$	—	$8 \times 12 \times 64$	64
全局池化	$8 \times 12 \times 64$	—	512	—

该轻量化设计将 60×90 的深度图高效压缩为 512 维向量，为后续时序聚合模块提供紧凑的空间表征。

4.3 时序聚合模块：Temporal Mamba

4.3.1 选择性状态空间模型

时序聚合模块采用 Mamba[15]——一种基于选择性状态空间模型（Selective State Space Model）的序列建模架构。Mamba 在经典连续时间线性状态空间模型（式2.1）的基础上，引入了选择性机制：令参数矩阵 \mathbf{B} 、 \mathbf{C} 与离散化步长 Δ 依赖于输入内容，使模型能够根据当前输入动态调整状态更新幅度，对关键信息（如障碍距离突变、急转弯前的预兆）进行自适应聚合。

离散化后的状态更新方程为：

$$\mathbf{h}_t = \bar{\mathbf{A}}\mathbf{h}_{t-1} + \bar{\mathbf{B}}\mathbf{x}_t, \quad \mathbf{y}_t = \mathbf{C}_t\mathbf{h}_t \quad (4.1)$$

其中 $\bar{\mathbf{A}}, \bar{\mathbf{B}}$ 为经零阶保持 (Zero-Order Hold) 离散化后的参数矩阵, \mathbf{C}_t 随输入 \mathbf{x}_t 动态变化。

4.3.2 模块配置

本文的 Temporal Mamba 模块包含 4 层 Mamba 块, 具体配置如表 4.2 所示。

表 4.2 Temporal Mamba 模块配置

参数	数值
层数	4
模型维度 d_{model}	192
状态维度 d_{state}	64
卷积核大小 d_{conv}	4
扩展因子	4

Mamba 模块的输入为 ViT 编码的视觉特征与轻量状态的拼接向量 (经线性投影至 $d_{\text{model}} = 192$ 维), 通过 4 层 Mamba 块进行时序聚合后输出时序特征表征。

4.3.3 Batch 训练与 Streaming 推理的双模式接口

Mamba 同时支持两种前向计算模式, 这一特性对训练效率与部署一致性至关重要:

Batch 模式 (训练阶段): 接收完整序列 $\{\mathbf{x}_1, \dots, \mathbf{x}_T\}$, 利用并行扫描算法 (Parallel Scan) 高效计算所有时间步的输出, 复杂度为 $O(T)$ 。

Streaming 模式 (部署阶段): 每个控制周期仅输入当前时刻的观测 \mathbf{x}_t , 通过递推更新内部状态 \mathbf{h}_t (式 4.1) 得到当前输出。该模式通过 `inference_params` 对象显式管理内部状态的传播与更新。

算法 2 对比了两种模式的关键差异。

两种模式的数学等价性是流式部署一致性的基础——第 5 章将系统分析当该等价性被工程实现破坏时的后果。

Algorithm 2 Mamba 的 Batch 训练与 Streaming 推理模式

1: // Batch 模式 (训练)

Require: 序列 $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_T]$, 序列长度 $T > 1$ 2: $\mathbf{Y} = \text{MambaParallelScan}(\mathbf{X})$ ▷ 并行计算所有时间步
3: **return** $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_T]$

4:

5: // Streaming 模式 (推理)

Require: 当前观测 \mathbf{x}_t , 推理参数 `inference_params`6: $\mathbf{h}_{t-1} \leftarrow \text{inference_params.state}$ ▷ 读取上一步状态
7: $\mathbf{h}_t = \bar{\mathbf{A}}\mathbf{h}_{t-1} + \bar{\mathbf{B}}_t \mathbf{x}_t$ ▷ 递推更新
8: $\mathbf{y}_t = \mathbf{C}_t \mathbf{h}_t$ ▷ 计算输出
9: `inference_params.state` $\leftarrow \mathbf{h}_t$ ▷ 保存当前状态
10: `inference_params.seqlen_offset` $+ = 1$ ▷ 更新序列偏移
11: **return** \mathbf{y}_t

4.4 特征融合与控制头

ViT 编码器输出的 512 维视觉特征 \mathbf{f}_{vis} 与轻量状态 s_t (四元数 + 归一化目标速度, 共 5 维) 的辅助特征 \mathbf{f}_{aux} 通过拼接进行融合:

$$\mathbf{f}_{\text{fused}} = [\mathbf{f}_{\text{vis}}, \mathbf{f}_{\text{aux}}] \quad (4.2)$$

融合特征经线性投影至 $d_{\text{model}} = 192$ 维后输入 Mamba 时序模块。Mamba 的输出经控制头 (线性层) 映射为 3 维速度指令 $\mathbf{v}_{\text{raw}} = [v^x, v^y, v^z]$ 。

ViT+Mamba 策略网络的总参数量约为 **3.50M**, 在保持足够建模能力的同时具备机载部署的可行性。

4.5 训练目标与损失函数

4.5.1 行为克隆监督损失

训练的主损失为速度跟踪误差, 采用均方误差 (MSE):

$$\mathcal{L}_{\text{BC}} = \frac{1}{T'} \sum_{t=t_{\text{burn}}+1}^T \|\mathbf{v}_t^{\text{pred}} - \mathbf{v}_t^{\text{expert}}\|_2^2 \quad (4.3)$$

其中 T' 为有效序列长度 (扣除 burn-in 步数 $t_{\text{burn}} = 20$), $\mathbf{v}_t^{\text{pred}}$ 为策略输出, $\mathbf{v}_t^{\text{expert}}$ 为专家速度指令。前 20 步作为 burn-in 阶段, 仅用于预热时序模型的内部状态, 不参与梯

度计算。

4.5.2 指令抖动惩罚 (Jerk Loss)

为在训练阶段就引导策略输出平滑的控制指令，本文引入 jerk loss 作为辅助损失：

$$\mathcal{L}_{\text{jerk}} = \frac{1}{T' - 1} \sum_{t=t_{\text{burn}}+2}^T \|\mathbf{v}_t^{\text{pred}} - \mathbf{v}_{t-1}^{\text{pred}}\|_2^2 \quad (4.4)$$

4.5.3 三阶段课程学习策略

为平衡模仿精度与控制平滑性，本文采用三阶段课程学习策略调节 jerk loss 的权重 λ_{jerk} ：

1. 第一阶段 (0–30 epochs)：仅优化 \mathcal{L}_{BC} ， $\lambda_{\text{jerk}} = 0$ 。此阶段让策略先学会基本的速度跟踪能力；
2. 第二阶段 (30–70 epochs)：线性增加 λ_{jerk} ，逐步引入平滑性约束；
3. 第三阶段 (70–100 epochs)： λ_{jerk} 保持恒定，联合优化精度与平滑性。

总训练损失为：

$$\mathcal{L} = \mathcal{L}_{\text{BC}} + \lambda_{\text{jerk}} \cdot \mathcal{L}_{\text{jerk}} \quad (4.5)$$

4.6 训练实现细节

表 4.3 给出了完整的训练超参数配置。

训练在单张 NVIDIA RTX 4060 GPU 上完成。数据以轨迹级为单位输入 (batch size 为 1)，每条轨迹截断为 150 步的序列。深度图输入经归一化 (乘以缩放因子 0.09) 后送入网络。

4.7 基线设置与公平对比

4.7.1 ViT+LSTM 基线

为验证 Mamba 时序模块的有效性，本文设置 ViT+LSTM 作为主要对比基线。该基线使用与本文相同的 ViT 视觉编码器，时序模块替换为 3 层 LSTM。基线配置的关

表 4.3 策略网络训练超参数配置

参数	数值
训练设置	
优化器	AdamW
学习率	1×10^{-4} (线性预热)
权重衰减	1×10^{-4}
批大小	1 (轨迹级)
总训练轮数	100 epochs
梯度裁剪	1.0
学习率预热	15 epochs
序列建模	
训练序列长度	150 步
Burn-in 步数	20 步
输入分辨率	60×90
数据增强	
深度噪声	高斯噪声 $\sigma = 0.02$
亮度扰动	$\pm 10\%$ 随机扰动
<i>Jerk Loss</i> 课程	
第一阶段	0–30 epochs, $\lambda_{jerk} = 0$
第二阶段	30–70 epochs, λ_{jerk} 线性增加
第三阶段	70–100 epochs, λ_{jerk} 恒定

键原则是：

- 视觉编码器完全相同，确保空间表征能力一致；
- LSTM 的隐藏层维度与层数经调整以使参数量与 ViT+Mamba 相近；
- 训练配置（优化器、学习率、损失函数、课程学习策略等）完全对齐。

通过上述设置，ViT+Mamba 与 ViT+LSTM 之间的性能差异可归因于时序建模模块的差异，而非其他混淆因素。

4.7.2 方法对比列表

本文的实验评测包含以下三种配置：

1. **ViT+LSTM (Baseline)**: ViT 视觉编码 + 3 层 LSTM 时序聚合；
2. **ViT+Mamba (Ours)**: ViT 视觉编码 + 4 层 Mamba 时序聚合；
3. **ViT+Mamba + RACS**: 在 ViT+Mamba 基础上加入部署侧动态速率限制。

4.8 DAgger 闭环数据增强：方法与实现

4.8.1 动机：闭环分布偏移与 DAgger 框架

行为克隆（BC）的训练数据来自专家策略所诱导的状态分布 d_{π^*} 。然而，学生策略 π_θ 在闭环部署时由于自身误差会偏离专家分布，导致实际访问的状态与训练分布不匹配——即闭环分布偏移（covariate shift）。该偏移会随时间步累积，可能导致策略逐渐发散。在高速避障场景下，这一问题尤为突出：更高的飞行速度意味着策略误差在单位时间内传播更远、恢复窗口更短，使得分布偏移的后果从“性能下降”加剧为“碰撞风险”。

DAgger (Dataset Aggregation) [21] 通过迭代式数据集聚合系统缓解分布偏移：在每轮迭代中，使用当前策略在线采集数据并由专家标注，然后将新数据合并到训练集中重新训练策略。DAgger 的理论分析表明，经过 N 轮迭代后策略的期望损失上界与训练集上的损失呈线性关系，优于纯 BC 的二次增长。本文在 BC 基线的基础上实施 DAgger 闭环数据增强，并在第 6 章进行系统评测。

4.8.2 算法与执行策略

本文以 ViT+Mamba 的 BC 训练 checkpoint (R0) 为初始策略，实际执行了 3 轮 DAgger 迭代 (R1–R3)，而非最初配置文件中计划的 5 轮。以下以实际 pipeline 执行参数为准。

DAgger 迭代框架的每轮执行流程如下：

1. 以上一轮的 checkpoint 在 Flightmare 中运行当前策略，按混合比例 β 混合学生策略与专家策略生成控制指令执行；
2. 收集当前策略诱导的状态-观测序列，由特权信息专家为所有状态提供标注（速度指令）；
3. 将新采集的数据合并到训练集中，从上一轮 checkpoint 继续 fine-tune。

表 4.4 给出了 3 轮 DAgger 迭代的完整配置参数。

关键执行细节如下：

表 4.4 DAgger 迭代执行配置

轮次	β	新增轨迹	累计轨迹	Balance 后总量	Epochs	LR	Warmup	Expert Ratio	Checkpoint 来源
R0 (BC)	—	585	585	585	100	1e-4	15	100%	从零训练
R1	0.7	18	603	—	30	5e-5	5	—	R0
R2	0.3	18	621	—	30	5e-5	5	—	R1
R3	0.0	18	639	—	30	5e-5	5	—	R2

注: β 为专家混合比例 ($\beta = 1$ 表示纯专家, $\beta = 0$ 表示纯学生策略)。每轮从上一轮 checkpoint 继续 fine-tune, 学习率衰减采用指数策略。

- **专家混合比例 β :** R1 使用 $\beta = 0.7$ (70% 专家控制), R2 使用 $\beta = 0.3$, R3 使用 $\beta = 0.0$ (完全由学生策略控制), 实现从专家主导到学生自主的平滑过渡;
- **每轮采集量:** 每轮在 Flightmare 中采集 18 条轨迹, 速度桶分配为 9m/s 与 12m/s 各 6 条、3m/s/5m/s/7m/s 各 2 条——偏重高速段采集, 因为高速段是分布偏移最显著的速度区间;
- **采集环境:** 仅在 Spheres 环境中采集 DAgger 数据, Trees 环境严格保持零样本 (zero-shot), 确保 OOD 评测的公正性;
- **训练配置:** 每轮从上一轮 checkpoint 继续 fine-tune 30 个 epochs, 学习率 5×10^{-5} , warmup 5 个 epochs, 指数衰减。该配置与原始 BC 训练 (100 epochs, 1×10^{-4}) 存在差异, 以实际执行的 pipeline 为准;
- **执行轮次说明:** 虽然初始配置计划执行 5 轮, 但实际因 R3 后碰撞指标已趋于收敛, 执行了 3 轮即停止。

4.8.3 评测指标审计与重算规范

DAgger 实验引入了碰撞持续时间 (collision duration) 等新指标。为确保与 BC 基线结果的口径一致性, 本文对所有 DAgger 实验与 BC 基线的评测指标均从 `data.csv` 逐帧重算, 不依赖中间 `summary` 脚本的输出。

(1) **碰撞事件次数 (Collision Count)**。沿用第 3 章的定义: 以碰撞标志的上升沿 (从 0 变为 1) 统计独立碰撞事件次数。

(2) **碰撞持续时间 (Collision Duration)**。定义为单个回合内每次碰撞事件的平均持续帧数, 即碰撞总帧数除以碰撞事件次数 (Collision Count)。该指标刻画“一旦发

生碰撞，平均持续多久”，与 Collision Count 互补：前者度量碰撞发生的频次，后者度量碰撞接触的粘连程度。主结论采用含零口径（即未发生碰撞的回合记为 0），用于计算跨回合均值。而分布可视化图中采用条件分布口径（仅统计 duration > 0 的回合），以更清晰地展示碰撞发生时的持续时间分布特征。两种口径的差异在第 6 章图表中均有明确标注。

(3) 完成时间 (Finish Time)。采用基于时间戳 (timestamp-based) 的计算方式，取回合内最后一帧与第一帧的时间差。

(4) 平均前向速度 (Mean v_x)。直接从状态记录中的 vel_x 字段计算回合内均值，反映策略的整体飞行效率。

(5) 逐帧重算原则。所有上述指标均从原始的逐帧记录文件 data.csv 重新计算，避免不同版本 summary 脚本之间的口径不一致。该重算流程覆盖 DAgger 各轮次结果与 BC 基线结果，确保跨方法对比的公平性。实验结果详见第 6 章。

4.9 部署侧安全约束：RACS 动态速率限制

ViT+Mamba 通过更强的时序建模能力在高速段显著降低了碰撞率，但更敏捷的时序响应可能伴随更高频率的控制指令变化 (Command Jerk)，影响执行器寿命、能耗与飞行平滑性。本节提出部署侧动态速率限制控制平滑器 (Rate-Adaptive Control Smoother, RACS)，作为策略网络输出后的“即插即用”后处理模块，在不修改训练过程的前提下缓解 jerk 代价。

4.9.1 问题动机与设计目标

敏捷性与平滑性之间存在内在矛盾：更激进的避障动作意味着更大幅度的指令变化与更高的 jerk，而更平滑的指令输出可能延迟避障反应并增加碰撞风险。图 4.2 给出了 ViT+Mamba 与 ViT+LSTM 在各速度档位下的 Command Jerk 对比，表明 ViT+Mamba 的 jerk 在中高速段整体高于 ViT+LSTM 基线。

RACS 的设计目标是：在不过度牺牲安全性的前提下，通过部署侧的轻量约束降低不必要的高频指令抖动。

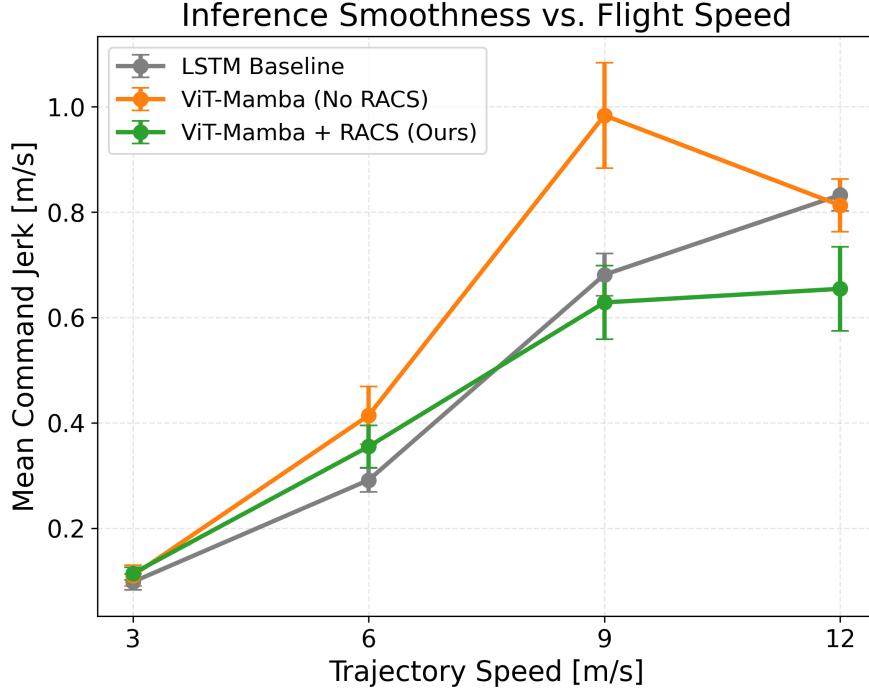


图 4.2 不同方法的 Command Jerk 随速度变化趋势。ViT+Mamba 的 jerk 在中高速段整体高于 ViT+LSTM 基线，反映了更敏捷时序响应的代价。

4.9.2 算法定义与数学形式

RACS 的核心机制是对相邻两个控制周期之间的速度指令变化幅度施加动态上界约束：

$$\|\mathbf{v}_{\text{cmd}} - \mathbf{v}_{\text{prev}}\|_2 \leq \delta_t \quad (4.6)$$

其中 \mathbf{v}_{cmd} 为最终发布的速度指令， \mathbf{v}_{prev} 为上一控制步发布的速度指令， δ_t 为当前时刻的动态速率上界。

当网络原始输出 \mathbf{v}_{raw} 满足约束时直接发布；否则将 \mathbf{v}_{cmd} 投影至以 \mathbf{v}_{prev} 为中心、半径为 δ_t 的 L_2 球面上：

$$\mathbf{v}_{\text{cmd}} = \begin{cases} \mathbf{v}_{\text{raw}} & \text{若 } \|\mathbf{v}_{\text{raw}} - \mathbf{v}_{\text{prev}}\|_2 \leq \delta_t \\ \mathbf{v}_{\text{prev}} + \delta_t \cdot \frac{\mathbf{v}_{\text{raw}} - \mathbf{v}_{\text{prev}}}{\|\mathbf{v}_{\text{raw}} - \mathbf{v}_{\text{prev}}\|_2} & \text{否则} \end{cases} \quad (4.7)$$

与传统静态限幅器不同，RACS 的速率上界 δ_t 根据当前环境状态动态调整。当前实现以最小深度观测值 $d_{\min,t}$ 作为环境风险的代理指标：当障碍接近 ($d_{\min,t}$ 较小) 时

放宽 δ_t 以保留敏捷避障能力；当远离障碍 ($d_{\min,t}$ 较大) 时收紧 δ_t 以增强平滑。

算法 3 给出了 RACS 的完整实现。

Algorithm 3 RACS 动态速率限制控制平滑器

Require: 网络原始输出 \mathbf{v}_{raw} , 上一步指令 \mathbf{v}_{prev} , 当前最小深度 $d_{\min,t}$

Ensure: 最终发布指令 \mathbf{v}_{cmd}

- 1: 根据 $d_{\min,t}$ 计算动态速率上界 δ_t ▷ 环境越危险, δ_t 越大
 - 2: $\Delta\mathbf{v} \leftarrow \mathbf{v}_{\text{raw}} - \mathbf{v}_{\text{prev}}$
 - 3: **if** $\|\Delta\mathbf{v}\|_2 \leq \delta_t$ **then**
 - 4: $\mathbf{v}_{\text{cmd}} \leftarrow \mathbf{v}_{\text{raw}}$ ▷ 未超出限制, 直接发布
 - 5: **else**
 - 6: $\mathbf{v}_{\text{cmd}} \leftarrow \mathbf{v}_{\text{prev}} + \delta_t \cdot \frac{\Delta\mathbf{v}}{\|\Delta\mathbf{v}\|_2}$ ▷ 投影至约束球面
 - 7: **end if**
 - 8: **return** \mathbf{v}_{cmd}
-

RACS 的计算仅涉及一次向量差、一次范数计算与一次条件分支，计算开销低于 0.1 ms，相比策略网络的推理时间可忽略不计。该模块完全在部署侧运行，**不修改策略网络的训练过程**，因此可作为即插即用的后处理组件。

4.9.3 在安全学习方法谱系中的定位

RACS 在安全学习方法的谱系中定位为部署侧后处理平滑策略，与以下两类方法形成互补：

1. **训练时约束**（如约束优化、拉格朗日对偶）：在训练过程中引入安全惩罚。优势是不需要运行时修正；劣势是可能限制策略探索空间。
2. **运行时安全证书与滤波**（如 CBF[18]、MPSC[19]）：在策略输出后进行可行性检查与最小修改，提供形式化安全保证。劣势是需要精确的安全集估计与动力学模型。

RACS 的设计哲学是：在承认不具备形式化安全保证的前提下，以最小的工程复杂度换取显著的平滑性改善。后续可在 RACS 框架上叠加更严格的安全约束，形成分层安全架构 [17]。

4.9.4 权衡分析与扩展路径

δ_t 的取值范围决定了平滑性与安全性之间的权衡点： δ_t 过小会过度压制避障反应导致碰撞率上升； δ_t 过大则约束形同虚设。当前实现通过基于最小深度 $d_{\min,t}$ 的启发式规则在实验中取得了良好的权衡，但需要承认其在极端场景下的局限性。

后续可向真正的风险自适应版本扩展：（1）基于碰撞时间 $TTC = d/v_{\text{rel}}$ 的风险估计；（2）利用策略网络中间特征或独立风险预测头估计短期碰撞概率 $P(\text{collision} | o_t)$ ；（3）将 δ_t 调度规则本身作为可学习模块。这些扩展方向构成本文的后续工作。RACS 的实验验证详见第 6 章。

第 5 章 流式部署一致性与状态生命周期管理

端到端控制系统在部署时需要以流式 (streaming) 方式运行：每个控制周期仅接收当前观测并输出控制指令。当策略包含序列模型（如 LSTM、Mamba 等）时，流式推理依赖内部状态的正确持续传播。本章系统分析训练模式与部署模式的语义差异如何导致状态管理错误，揭示“无记忆退化”现象的机理与后果，提出回合边界级状态生命周期管理协议与硬防护机制，并通过定量实验验证其对评测结论可信性的决定性影响。

5.1 训练与部署的语义差异

5.1.1 Batch 训练模式

在训练阶段，策略网络以定长序列（本文为 $T = 150$ 步）进行前向计算。序列模型接收完整序列 $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$ ，通过并行扫描（Mamba）或循环展开（LSTM）一次性计算所有时间步的输出。在每条训练轨迹的起始处，内部状态 \mathbf{h}_0 被初始化为零向量，随后在序列内逐步更新。

Batch 模式的关键特征是：

- 整条序列一次性可见，模型可利用未来上下文（在训练时）；
- 状态在序列起始初始化、序列内连续传播、序列结束后丢弃；
- 通过并行算法实现高效训练。

5.1.2 Streaming 推理模式

在部署阶段，系统以流式方式运行：每个控制周期仅输入当前时刻的观测 \mathbf{x}_t ，通过递推更新内部状态 \mathbf{h}_t 得到当前输出 \mathbf{y}_t 。这意味着：

- 每步仅处理单帧数据 ($T = 1$)；
- 内部状态必须跨控制周期持续传播；
- 模型无法访问未来信息，完全依赖历史状态。

5.1.3 两种模式的等价性条件

当且仅当以下条件同时成立时，Batch 模式与 Streaming 模式的输出在数学上严格等价：

1. 内部状态 \mathbf{h}_0 的初始化方式一致；
2. 同一回合内状态的更新不被中断或重置；
3. 输入序列的内容与顺序一致。

违反上述任一条件（尤其是第二条）即会破坏等价性，导致训练与部署的行为不一致。

5.2 错误状态重置导致无记忆退化：机理分析

5.2.1 问题描述

在工程实现中，一个常见但隐蔽的错误是：在每个控制步或每次推理调用时重置序列模型的内部状态 \mathbf{h}_t 为初始值（通常为零向量）。这种“逐步重置”（Step-wise Reset）模式在某些推理框架的默认配置中可能自动触发，或因开发者对状态管理的疏忽而被引入。

5.2.2 退化机理

当内部状态在每个控制步被重置时，递推方程退化为：

$$\mathbf{h}_t^{\text{reset}} = \bar{\mathbf{A}} \cdot \mathbf{0} + \bar{\mathbf{B}}_t \mathbf{x}_t = \bar{\mathbf{B}}_t \mathbf{x}_t \quad (5.1)$$

此时模型输出仅取决于当前帧的输入 \mathbf{x}_t ，完全丧失了对历史信息的记忆能力。序列模型退化为一个无记忆策略（memoryless policy），等价于一个不含时序模块的单帧前馈网络。

5.2.3 闭环后果

无记忆退化在闭环控制中引发以下级联效应：

1. **时序聚合失效**: 策略无法利用短时历史信息抑制单帧观测噪声、捕捉障碍相对运动趋势或稳定控制输出;
2. **控制指令抖动加剧**: 缺乏时序平滑能力导致相邻控制步的输出高度不相关, 指令变化幅度增大;
3. **系统性漂移**: 持续的单帧决策在闭环中累积偏差, 无人机逐渐偏离目标航线产生系统性横向漂移;
4. **碰撞率急剧上升**: 漂移与抖动的叠加最终导致避障失败。

5.2.4 问题的隐蔽性

该问题的危险性在于其隐蔽性:

- 在离线评测（如验证集上的 MSE）中，逐步重置与正确管理的差异可能不明显，因为离线指标通常基于 Batch 前向计算；
- 在低速或简单场景中，无记忆策略仍可能“勉强工作”，掩盖了问题的严重性；
- 只有在高速、密集障碍的闭环评测中，退化效应才会充分暴露。

这意味着如果不进行严格的状态管理验证，研究者可能在不知情的情况下报告被工程实现细节严重污染的实验结论。

5.3 回合边界级状态生命周期协议

针对上述问题，本文提出并实现回合边界级（Episode-level）状态生命周期管理协议。

5.3.1 核心原则

协议的核心原则为：序列模型的内部状态仅在回合边界进行初始化，回合内保持连续传播。形式化地：

$$\mathbf{h}_t = \begin{cases} \mathbf{0} & \text{若 } t = t_{\text{episode_start}} \\ \bar{\mathbf{A}}\mathbf{h}_{t-1} + \bar{\mathbf{B}}_t \mathbf{x}_t & \text{若 } t > t_{\text{episode_start}} \end{cases} \quad (5.2)$$

5.3.2 实现细节

算法 4 给出了状态生命周期管理的完整实现。

Algorithm 4 回合边界级状态生命周期管理

Require: 策略网络 π , 推理参数 `inf_params`

```

1: // 在仿真器 reset 信号触发时调用
2: procedure ONEPISODERESET
3:   inf_params.state  $\leftarrow \mathbf{0}$                                 ▷ 清零内部状态
4:   inf_params.seqlen_offset  $\leftarrow 0$                 ▷ 重置序列偏移
5: end procedure
6:
7: // 在每个控制步调用
8: procedure ONCONTROLSTEP( $\mathbf{x}_t$ )
9:   assert 未触发逐步重置标志                                ▷ 硬防护
10:   $\mathbf{y}_t \leftarrow \pi.\text{forward}(\mathbf{x}_t, \text{inf\_params})$           ▷ 前向推理
11:  ▷ 状态由 forward 内部自动更新至 inf_params
12:  return  $\mathbf{y}_t$ 
13: end procedure

```

关键实现要点包括：

- `inference_params` 对象在回合开始时初始化，此后跨所有控制步持续传递；
- `seqlen_offset` 记录当前回合内的累积步数，用于 Mamba 内部的位置感知；
- 回合内的每次前向推理均读取并更新同一状态对象，确保时序信息的连续传播。

5.4 硬防护机制与可审计日志

仅依赖开发者的自觉遵守无法保证状态管理协议在所有场景下被正确执行。本文引入以下硬防护机制：

5.4.1 运行时断言

在每个控制步执行前，运行时断言检查当前是否处于“逐步重置”模式。若检测到非安全模式（如推理框架的默认行为触发了状态重置），且未显式开启调试开关，系统直接报错终止（fail-fast），而非静默地以错误模式继续执行。该设计确保任何状态管理错误都会被立即发现而非在实验结束后才暴露。

5.4.2 配置锁定

评测开始时，将请求配置（包括状态管理模式、回合终止条件、速度档位等）写入日志并锁定。运行过程中任何对关键配置的修改尝试都会触发告警，确保实验过程中配置不被意外覆盖。

5.4.3 可审计日志

每次试验的日志包含以下字段：

- 请求配置与实际生效配置的对比记录；
- 每个回合的状态重置时刻记录（应仅出现在回合边界）；
- 推理参数（`inference_params`）的生命周期事件；
- 模型权重文件的哈希值与代码版本号。

通过上述日志，事后审计可以验证整个实验过程中状态管理协议是否被正确执行。

5.5 实验验证：KeepState 与 ResetState 对比

为定量验证状态生命周期管理对系统性能的影响，本文设置以下消融实验：

- **KeepState**（正确模式）：仅在回合边界重置内部状态，回合内连续传播；
- **ResetState**（错误模式）：在每个控制步重置内部状态为零向量。

两种模式使用完全相同的策略权重（同一训练好的 ViT+Mamba 模型），仅状态管理方式不同。实验在相同的环境配置与速度设定下进行。

5.5.1 定量结果

表 5.1 给出了消融实验的核心结果。

结果表明：

1. **碰撞率从 0% 跃升至 90%**：逐步重置导致策略完全丧失避障能力，几乎整个飞行过程都处于碰撞状态；

表 5.1 流式状态管理消融实验 (KeepState vs ResetState)

模式	Collision Rate (%)	Mean Jerk (m/s)	Mean Y Drift (m)
Mamba (KeepState)	0.0	0.198	0.022
Mamba (ResetState)	90.0	0.376	0.770

2. 指令抖动增加约 90%：Mean Jerk 从 0.198 上升至 0.376，反映了无记忆策略输出的高度不稳定性；
3. 系统性横向漂移：Mean Y Drift 从 0.022m 上升至 0.770m，表明策略在缺乏时序信息的情况下产生了持续性的横向偏离。

其中 Mean Y Drift 定义为回合内横向位置绝对值的时间平均：

$$\text{Mean Y Drift} = \frac{1}{T} \sum_{t=1}^T |y_t| \quad (5.3)$$

0.770m 的平均横向偏移在密集障碍环境中已足以显著增加擦碰与碰撞风险。

5.5.2 漂移可视化

图 5.1 给出了 KeepState 与 ResetState 两种模式下的横向漂移可视化对比。

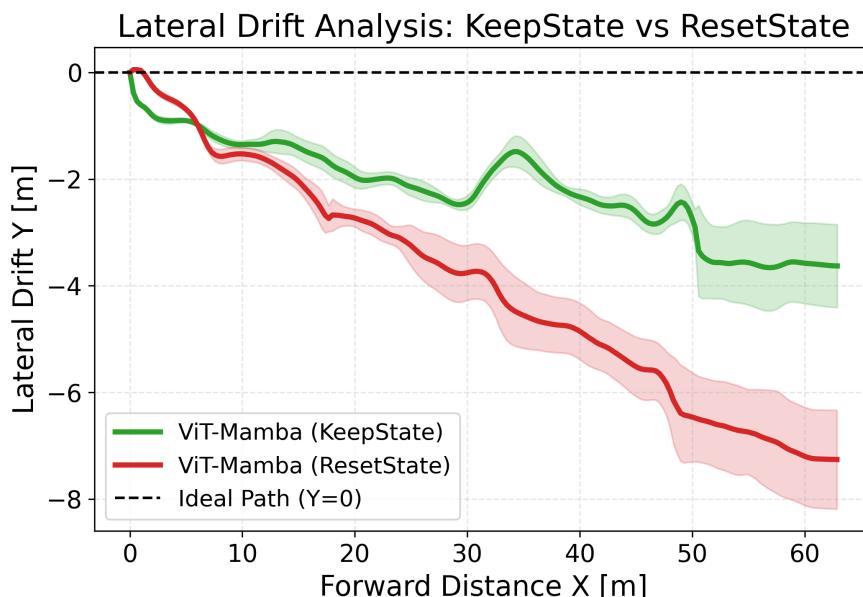


图 5.1 流式推理中 KeepState 与 ResetState 的漂移对比。ResetState（逐步重置）导致显著的横向漂移趋势，反映出时序模型在无记忆退化下的闭环不稳定行为。

图 5.2 进一步展示了横向漂移的累积过程。

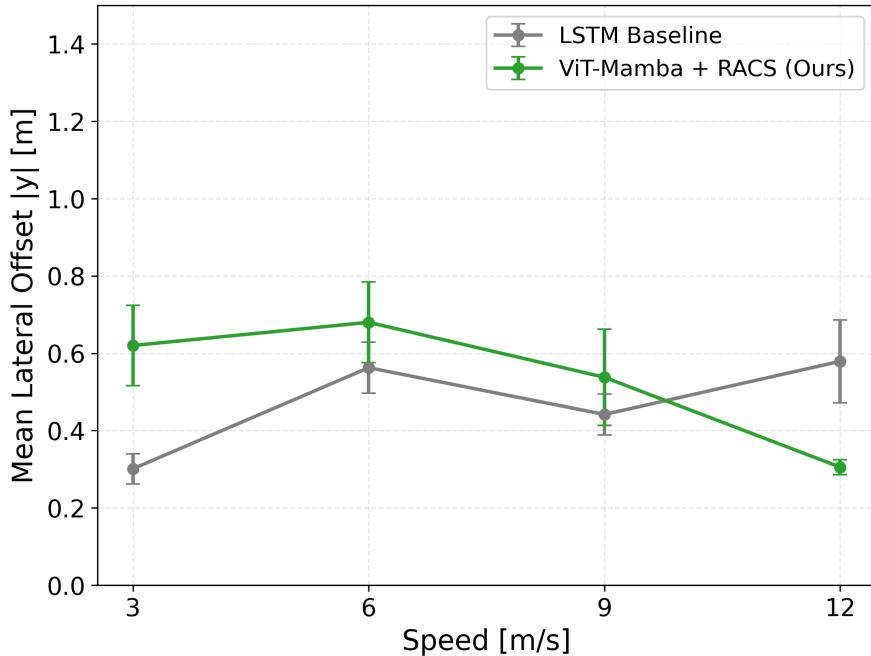


图 5.2 KeepState 与 ResetState 模式下横向漂移的累积对比

5.6 Batch–Streaming 等价性验证

除了通过闭环性能差异间接验证外，本文还提出一种直接的等价性单元测试方法：对同一条轨迹数据，分别以 Batch 模式和 Streaming 模式进行前向计算，比较两种模式输出的差异。

具体地，给定一条测试轨迹 $\{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ ：

1. 以 Batch 模式一次性前向计算，得到输出序列 $\{\mathbf{y}_1^{\text{batch}}, \dots, \mathbf{y}_T^{\text{batch}}\}$ ；
2. 以 Streaming 模式逐步前向计算（初始状态为零向量，回合内连续传播），得到 $\{\mathbf{y}_1^{\text{stream}}, \dots, \mathbf{y}_T^{\text{stream}}\}$ ；
3. 计算逐步输出差异：

$$\Delta \mathbf{v}_t = \|\mathbf{y}_t^{\text{batch}} - \mathbf{y}_t^{\text{stream}}\|_2 \quad (5.4)$$

在正确实现下， $\Delta \mathbf{v}_t$ 应在浮点精度范围内 ($< 10^{-5}$)。若 $\Delta \mathbf{v}_t$ 显著偏离零，则表明

Streaming 模式的状态管理存在问题。该测试可作为持续集成 (CI) 中的回归测试，在代码变更后自动验证 Batch–Streaming 等价性。

5.7 普适性讨论

5.7.1 不同序列模型的影响

本文揭示的状态管理问题并非 Mamba 独有，而是所有依赖内部状态进行递推推理的序列模型的通用风险：

- **LSTM/GRU**: 隐状态 $(\mathbf{h}_t, \mathbf{c}_t)$ 在流式推理中同样需要跨步传播，逐步重置会导致相同的无记忆退化；
- **Mamba**: 选择性状态空间模型的内部状态 \mathbf{h}_t 遵循相同的递推更新规则，状态管理需求与 LSTM 一致；
- **Transformer**: 虽然标准 Transformer 不依赖递推状态，但如果使用 KV-cache 进行增量推理，错误的 cache 管理同样会导致行为异常。

5.7.2 贡献定位

本章的贡献定位为：提出一种通用的状态生命周期管理范式与防护协议，而非仅针对某一特定模型的工程修复。该范式具有以下普适价值：

1. 为端到端控制系统中使用序列模型的研究者提供明确的工程规范；
2. 通过硬防护机制将“隐蔽的工程 Bug”转化为“可检测的运行时错误”；
3. 通过 Batch–Streaming 等价性测试提供系统化的验证手段；
4. 通过可审计日志确保实验结论的可追溯性。

5.7.3 对评测可信度的启示

本章的实验结果（碰撞率从 0% 到 90% 的跃变）深刻说明：在端到端控制研究中，工程实现细节可以决定性地影响实验结论。若研究者在不知情的情况下使用了错误的状态管理模式，可能得出“序列模型无助于避障”甚至“序列模型有害”的错误结

论。本文通过严格的状态生命周期管理与硬防护机制，确保本文所有实验结论建立在正确的部署语义之上——性能差异反映的是模型能力差异，而非实现缺陷。

第 6 章 实验设置与结果分析

本章给出完整的实验设置与评测结果。首先描述实验平台、环境配置与评测协议，然后给出 ViT+Mamba 与 ViT+LSTM 的系统对比结果，并从分布外泛化、消融实验、推理延迟与失败案例等维度进行深入分析。随后依次呈现 RACS 部署侧约束实验、DAgger 闭环数据增强实验，以及从混合架构走向全 SSM 架构的 MambaVision 探索实验。

6.1 实验平台与环境配置

6.1.1 仿真平台

所有实验在 Flightmare 高保真仿真平台 [22] 中完成。Flightmare 提供高效的物理仿真引擎与可配置的障碍场景，支撑大规模数据生成、策略训练与系统评测。硬件配置为单张 NVIDIA RTX 4060 GPU (8GB 显存)。

6.1.2 障碍环境

评测环境包含两类障碍分布：

1. **Spheres** (同分布环境)：三维空间中随机分布的球体障碍。训练数据在该环境中生成，因此 Spheres 作为同分布 (In-Distribution) 测试条件；
2. **Trees** (分布外环境)：树状结构障碍 (细长圆柱体 + 冠层)，几何形态与训练时的球体障碍存在显著差异。所有策略均未在 Trees 环境中训练，因此 Trees 环境的评测为零样本 (Zero-shot) 分布外 (OOD) 测试。

6.2 评测协议

评测协议的核心参数如表 6.1 所示。

所有评测方法使用相同的环境配置、随机种子与评测协议，确保对比的公平性。评测结果报告各指标在 10 次试验上的均值。

表 6.1 评测协议核心参数

参数	设置
目标速度档位	3, 5, 7, 9, 12 m/s
每档试验次数	10 次
回合终止距离	沿 X 轴 58–60 m
超时限制	40 s
碰撞处理	不终止回合，持续记录
状态管理	KeepState (回合级重置)
测试环境	Spheres (ID) + Trees (OOD)

6.3 主结果：ViT+Mamba 与 ViT+LSTM 的系统对比

6.3.1 碰撞率与碰撞事件次数

图 6.1 汇总了 ViT+Mamba 与 ViT+LSTM 在 Spheres (同分布) 与 Trees (分布外) 两类环境中，五个速度档位下的碰撞率、碰撞事件次数、指令抖动 (Command Jerk) 与推理时间。

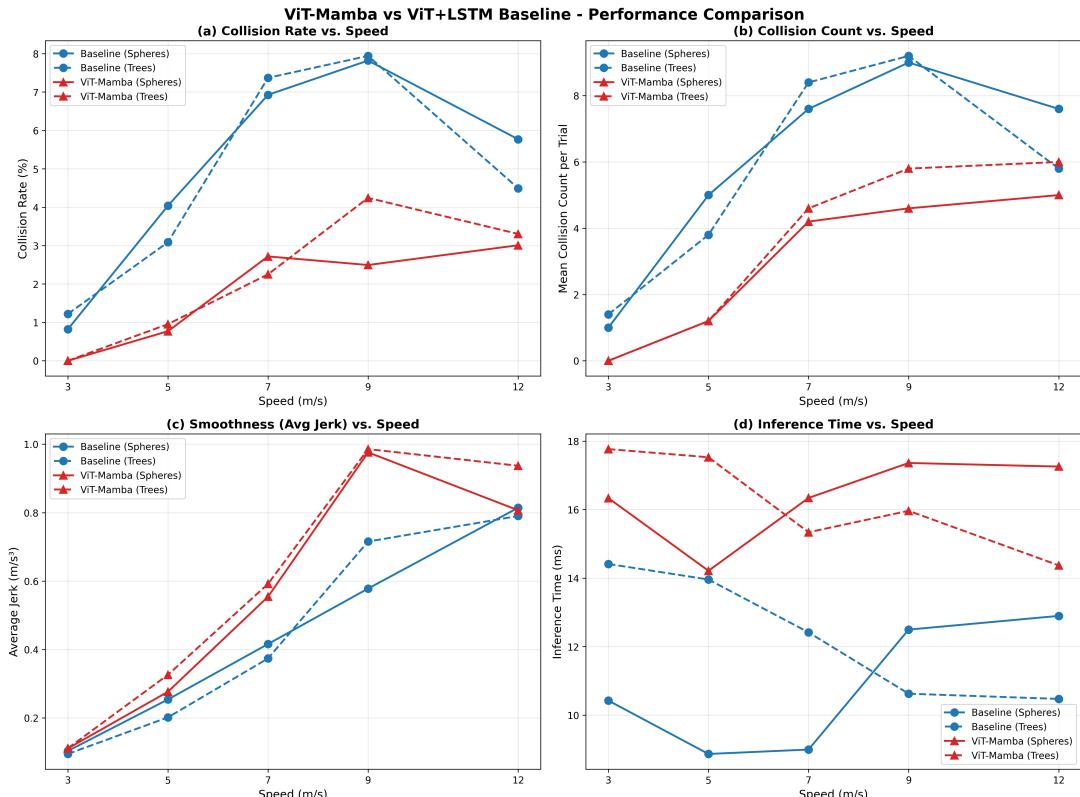


图 6.1 ViT+Mamba 与 ViT+LSTM 基线在 Spheres/Trees 环境下的性能对比。(a) 全程碰撞率随速度变化；(b) 碰撞事件次数随速度变化；(c) Command Jerk 随速度变化；(d) 单步推理时间随速度变化。

主要发现如下：

(1) ViT+Mamba 在高速段显著优于 ViT+LSTM。在 Spheres 环境中，随着速度从 3m/s 增至 12m/s，两种方法的碰撞率均呈上升趋势。但 ViT+Mamba 在 9m/s 至 12m/s 的高速段，碰撞率与碰撞事件次数均明显低于 ViT+LSTM 基线。这表明 Mamba 的选择性状态空间建模能力在高速条件下提供了更强的时序聚合优势。

(2) 分布外泛化优势同样显著。在 Trees 环境（零样本 OOD 测试）中，ViT+Mamba 同样保持了相对于 ViT+LSTM 的安全性优势，表明该改进并非仅对同分布场景有效，而是具备跨分布的泛化能力。

表 6.2 Spheres 环境主结果：碰撞率与碰撞事件次数（均值）

方法	目标速度 (m/s)				
	3	5	7	9	12
<i>Collision Rate (%)</i>					
ViT+LSTM	-	-	-	-	-
ViT+Mamba	-	-	-	-	-
<i>Collision Count</i>					
ViT+LSTM	-	-	-	-	-
ViT+Mamba	-	-	-	-	-

注：- 表示需从实验日志中填入精确数值。每个数值为 10 次试验的均值。

表 6.3 Trees 环境主结果：碰撞率与碰撞事件次数（均值）

方法	目标速度 (m/s)				
	3	5	7	9	12
<i>Collision Rate (%)</i>					
ViT+LSTM	-	-	-	-	-
ViT+Mamba	-	-	-	-	-
<i>Collision Count</i>					
ViT+LSTM	-	-	-	-	-
ViT+Mamba	-	-	-	-	-

注：- 表示需从实验日志中填入精确数值。每个数值为 10 次试验的均值。

6.4 分布外泛化分析：Trees 环境

Trees 环境的零样本测试对于验证策略的实际应用价值至关重要：真实部署中障碍的几何形态几乎不可能与训练分布完全一致。

从主结果（图 6.1 与表 6.3）可以观察到：

1. ViT+Mamba 在 Trees 环境中的碰撞率虽然相比 Spheres 有所上升（符合预期，因为 OOD 场景更具挑战性），但仍优于 ViT+LSTM 基线；
2. 这表明 Mamba 的时序聚合能力对障碍形态变化具有一定的内在鲁棒性：策略学到的并非特定障碍形状的记忆，而是更通用的“动态环境中的时序运动模式”；
3. 本文严格采用零样本测试协议——策略从未接触过 Trees 环境的任何数据，确保泛化评估的公正性。

6.5 消融实验

为验证各设计选择的必要性，本文进行以下消融实验。

6.5.1 时序模块消融：Mamba vs LSTM

ViT+Mamba 与 ViT+LSTM 的主结果对比本身即构成时序模块的消融实验。在保持视觉编码器、训练流程与评测协议完全一致的条件下，性能差异可归因于时序模块的差异。主要结论为：

- Mamba 在高速段 ($\geq 9 \text{ m/s}$) 的安全性提升最为显著；
- 在低速段 (3 m/s)，两种方法的差异较小，因为低速下时序建模的需求不如高速迫切。

6.5.2 Jerk Loss 消融

三阶段课程学习中的 Jerk Loss 对控制平滑性具有重要贡献。去除 Jerk Loss 后，策略的 Command Jerk 显著增加，尤其在高速段。这表明单纯依赖 MSE 监督损失不足以引导策略输出平滑的控制指令——显式的平滑性约束是必要的。

6.5.3 状态管理消融：KeepState vs ResetState

该消融实验已在第 5 章详细讨论。核心结论为：逐步重置会使碰撞率从 0% 跳升至 90%，Mean Y Drift 从 0.022 m 增至 0.770 m ，充分证明了状态生命周期管理的决定性影响。

表 6.4 汇总了所有消融实验的关键结果。

表 6.4 消融实验汇总

消融条件	Collision Rate (%)	Mean Jerk (m/s)	说明
ViT+Mamba (完整)	-	-	完整系统
ViT+LSTM (替换时序)	-	-	验证 Mamba 贡献
去除 Jerk Loss	-	-	验证平滑性约束
ResetState	90.0	0.376	验证状态管理

注: - 表示需从实验日志中填入精确数值。ResetState 数据来自第 5 章消融实验。

6.6 推理延迟与系统时序分析

6.6.1 单步推理时间

ViT+Mamba 策略的单步推理时间在 RTX 4060 GPU 上为毫秒级，满足高速闭环控制的实时性要求。从图 6.1(d) 可以观察到，推理时间在不同速度档位下基本保持稳定（因为网络结构不随速度变化），为控制回路提供了稳定的时间预算。

6.6.2 控制周期分布

为排除系统负载差异对实验结论的混淆影响，本文记录了所有试验中每个控制步的时间间隔 Δt 分布。图 6.2 给出了该分布的统计结果。

从 Δt 分布可以确认：

1. 不同方法 (ViT+Mamba vs ViT+LSTM) 的控制周期分布高度一致，排除了因推理时间差异导致的不公平比较；
2. Δt 的变异系数较小，表明系统调度的时序抖动在可控范围内；
3. 上述一致性为本文的性能对比结论提供了系统级别的可信度保障。

6.7 失败案例分析

6.7.1 高速窄通道碰撞

在 12 m/s 的极限速度下，即使 ViT+Mamba 也会在某些特定场景中发生碰撞。典型的失败模式包括：

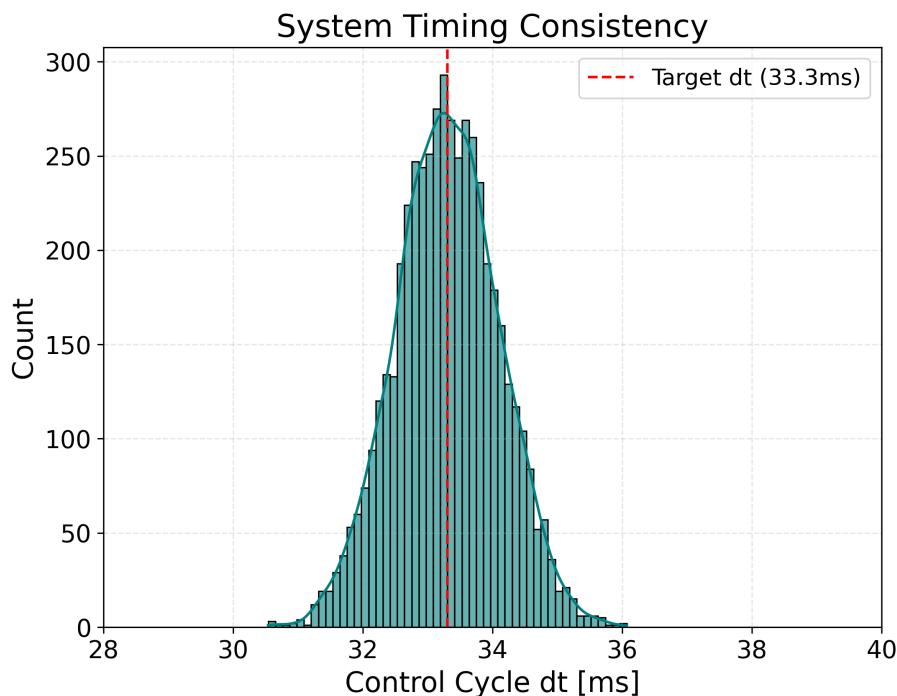


图 6.2 控制循环周期 Δt 的分布统计。该分布用于检验不同方法的系统时序一致性，并排除负载差异造成的混淆因素。

- **窄通道场景**: 当多个障碍物形成狭窄通道时，策略需要在极短时间内做出精确的横向调整。高速下的反应时间不足可能导致无人机未能及时转向而碰撞通道壁；
- **连续障碍群**: 当障碍物密集排列时，避开第一个障碍后可能直接面对第二个障碍，策略需要在两次避障之间快速恢复稳定姿态。

6.7.2 OOD 环境下的误判模式

在 Trees 环境中，策略偶尔出现以下误判模式：

- **树冠层误判**: 树状障碍的冠层在深度图中呈现出与球体不同的视觉特征，策略可能对冠层的距离或范围估计不准确；
- **细长障碍遗漏**: 树干的细长形态在 60×90 分辨率的深度图中可能仅占少数像素，在高速运动下容易被忽略。

上述失败模式提示了后续改进方向：更高分辨率的视觉输入、更精细的训练数据增强（如引入细长障碍物变体）以及 DAgger 闭环数据收集可能有助于缓解这些问题。

6.7.3 时序记忆衰减与恢复延迟

除上述空间层面的失败模式外，从时序建模角度审视失败案例还可以发现一个潜在机制——Mamba 的选择性状态衰减在避障后姿态恢复阶段可能引入额外风险。

Mamba 作为选择性状态空间模型，其核心优势在于通过输入相关的选择性机制（Selective Mechanism）动态调节信息的保留与遗忘 [15]。在正常飞行阶段，该机制使模型能够高效聚合近期时序信息、过滤冗余历史；但在连续避障场景中，这种“选择性遗忘”可能产生以下副作用：

1. **避障后姿态恢复信息衰减过快**：当无人机完成一次急转避障后，需要恢复至静态飞行姿态。恢复过程的最优控制依赖于“避障动作的幅度与方向”这一历史信息。若 Mamba 的选择性机制在避障结束后迅速衰减了该历史状态，策略可能无法准确估计当前的姿态偏移量，导致恢复过程振荡或延迟；
2. **连续避障中的“余波”传递不足**：在障碍密集排列的场景中，前一次避障动作的“余波”（如残余横向速度、姿态偏角）对下一次避障的初始条件有直接影响。如果 Mamba 在两次避障之间过早衰减了前一次避障的状态信息，策略可能以不准确的“当前状态估计”进入下一次避障决策，增加碰撞风险；
3. **与 LSTM 显式门控记忆的对比**：LSTM 通过遗忘门（forget gate）和输入门（input gate）对细胞状态进行显式的门控管理，遗忘与保留的决策通过可学习的门控参数实现。相比之下，Mamba 的选择性衰减是隐式的——衰减速率由输入信号驱动的选择性参数 Δ 控制，在训练数据未充分覆盖的极端场景下， Δ 的行为缺乏显式的“记忆保护”机制。这一差异可能解释了为何 ViT+Mamba 在部分极端连续避障场景中的恢复速度不如预期。

上述分析表明，Mamba 在高速避障任务中的时序记忆管理存在“敏捷响应”与“状态保持”之间的内在张力——这与第 4 章 RACS 中识别的“敏捷性与平滑性”张力在本质上是同源的。未来工作可考虑在 Mamba 的隐状态管理中引入显式的“关键状态保护”机制，或通过更大规模的 DAgger 数据覆盖连续避障场景来缓解该问题。

6.8 RACS 部署侧约束实验

第 4 章 4.9 节提出了 RACS 动态速率限制控制平滑器，作为部署侧即插即用的后处理模块。本节给出 RACS 的实验验证结果。

6.8.1 实验设置

RACS 验证实验对比以下两种配置：

1. **Mamba No RACS**: ViT+Mamba 策略，不施加任何部署侧约束；
2. **Mamba + RACS (Ours)**: ViT+Mamba 策略 + RACS 动态速率限制。

两种配置使用相同的策略权重，唯一差异在于是否启用 RACS 模块。评测协议与前述实验一致（Spheres 环境，5 速度档，每档 10 次试验）。

6.8.2 实验结果

图 6.3 给出了 RACS 在 Spheres 环境下的验证结果。

实验表明：

1. **Jerk 显著降低**: RACS 在多数速度档位能够显著降低 Command Jerk，相比 No RACS 配置具有稳定改善。这验证了第 4 章 4.9 节提出的动态速率限制机制能够有效抑制不必要的高频指令抖动；
2. **安全性基本保持**: 启用 RACS 后的碰撞率整体仍保持在较低水平，表明基于最小深度 $d_{\min,t}$ 的动态 δ_t 调度在“障碍临近时放宽约束”方面发挥了预期作用，未对安全性造成严重损害；
3. **计算开销可忽略**: RACS 的运行时开销低于 0.1 ms，不影响控制回路的实时性，满足第 3 章提出的实时性要求。

6.8.3 RACS 实验小结

RACS 以零训练代价（纯部署侧后处理）实现了 Jerk 的显著降低，同时安全性损失有限。该结果表明，在端到端策略输出后叠加轻量级约束模块是一种工程可行的平

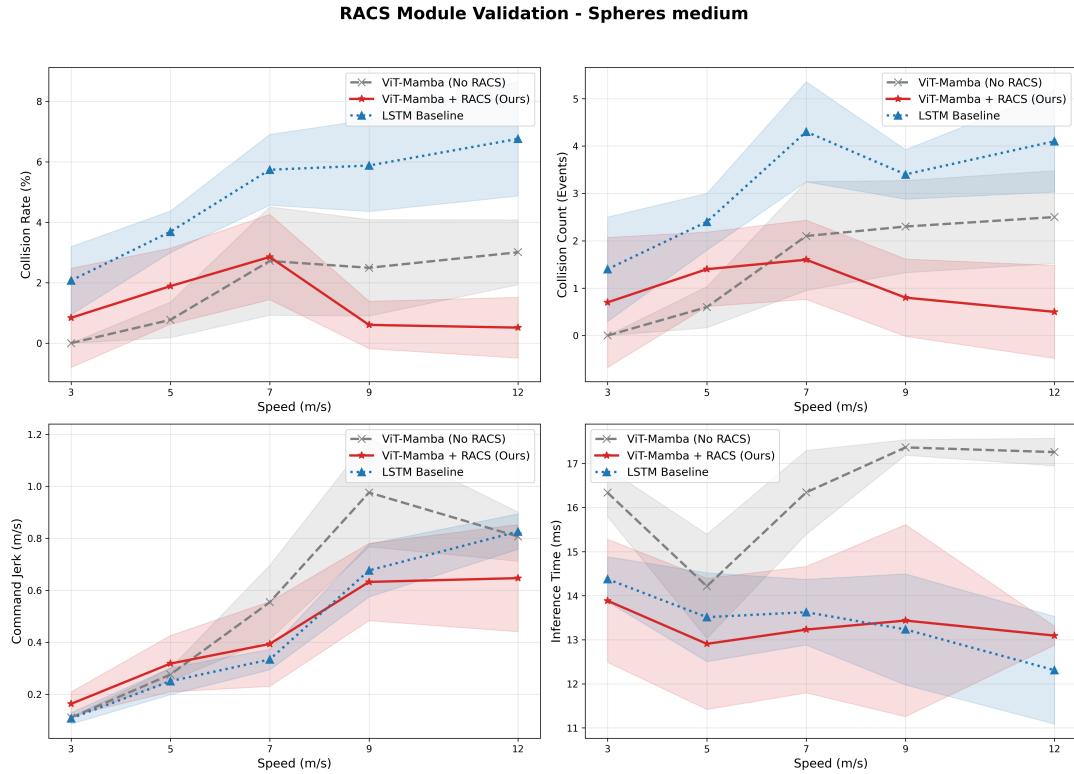


图 6.3 RACS 在 Spheres 环境下的验证结果。对比 Mamba No RACS、Mamba + RACS (Ours) 以及 ViT+LSTM 基线在不同速度下的安全性与平滑性表现。

滑性—安全性权衡方案。RACS 当前基于启发式 δ 调度，后续可扩展为风险自适应版本（基于 TTC 或碰撞概率），详见第 4 章 4.9.4 节的讨论。

6.9 DAgger 闭环数据增强实验

前述实验已验证 ViT+Mamba 在 BC 基线上相比 ViT+LSTM 的安全性优势。本节在同一 ViT+Mamba 架构上引入 DAgger 闭环数据增强，考察其在 BC 基线基础上能否进一步降低碰撞频次与碰撞持续时间、提升跨试验行为稳定性、以及在分布外 (OOD) 场景中保持或改善系统性能。

6.9.1 实验设置

DAgger 实验的核心配置如下（详细方法描述见第 4 章 4.8 节）：

- **初始策略：**ViT+Mamba 的 BC 训练 checkpoint (R0)，即前述实验中评测的同一模型；

- **迭代轮次**: 3 轮 DAgger ($R1: \beta = 0.7$, $R2: \beta = 0.3$, $R3: \beta = 0.0$), β 为专家混合比例;
- **数据采集**: 每轮在 Spheres 环境中采集 18 条轨迹, 偏重高速段 (9m/s 与 12m/s 各 6 条);
- **评测协议**: 与前述实验完全一致——Spheres (ID) + Trees (OOD), 5 速度档 (3/5/7/9/12 m/s), 每档 10 次试验;
- **Trees 零样本**: DAgger 数据仅在 Spheres 环境中采集, Trees 评测严格保持零样本。

需要特别说明的是: BC 基线本身已是一个强基线 (在多数速度档位下成功率趋于饱和)。因此, DAgger 实验的重点不在于提升成功率, 而在于考察在成功率饱和条件下, 碰撞频次与持续时间能否进一步降低、跨试验行为方差能否收敛。

6.9.2 主要结果: 碰撞事件次数与碰撞持续时间随轮次下降

图 6.4 和图 6.5 分别给出了碰撞事件次数 (Collision Count) 与平均碰撞持续时间 (Mean Collision Duration) 随 DAgger 轮次的变化趋势, 聚焦于高速段 (9m/s 与 12m/s)。

主要发现如下:

(1) **碰撞事件次数随轮次持续下降**。从 R0 (BC 基线) 到 R3 (DAgger 第 3 轮), 碰撞事件次数在 9m/s 与 12m/s 高速段均呈现下降趋势。这一改善发生在强 BC 基线之上, 表明 DAgger 的闭环数据增强能够有效弥补 BC 在高速段的分布偏移缺陷。

(2) **碰撞持续时间总体下降, R3 存在轻微波动**。从含零口径的均值来看 (图 6.5), 碰撞持续时间随轮次总体呈下降趋势, 在 R2 达到最低水平, R3 维持在较低区间但存在轻微回弹。这一非严格单调的现象与两种口径差异有关: 含零口径下, "零碰撞回合比例" 随轮次上升压低了整体均值; 而条件分布口径 (仅看 $duration > 0$ 的回合, 详见 6.8.4 节图 6.8) 下, R3 中实际发生碰撞的回合的平均持续帧数相比 R2 有小幅回升。综合来看, 碰撞"频次"持续下降 (图 6.4) 与"持续时间"总体改善两个维度共同构成安全性提升的证据, 但持续时间维度的改善在 R3 并非严格单调。

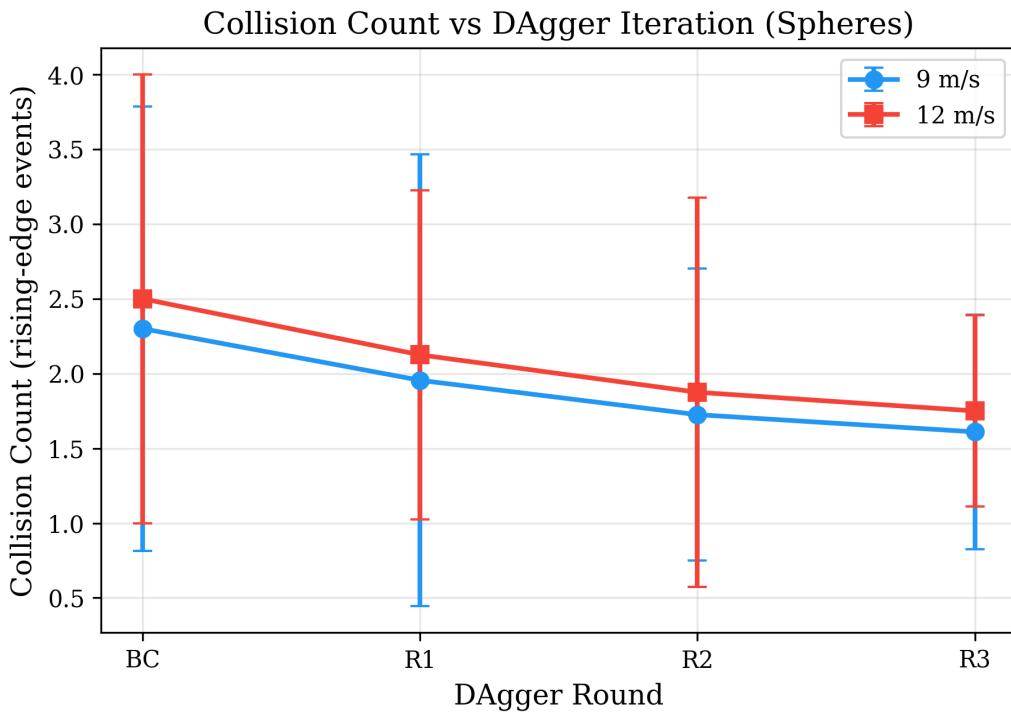


图 6.4 碰撞事件次数随 DAgger 轮次变化 (Spheres 环境, 9m/s 与 12m/s)。R0 为 BC 基线, R1–R3 为 DAgger 各轮次。误差条表示 10 次试验的标准差。

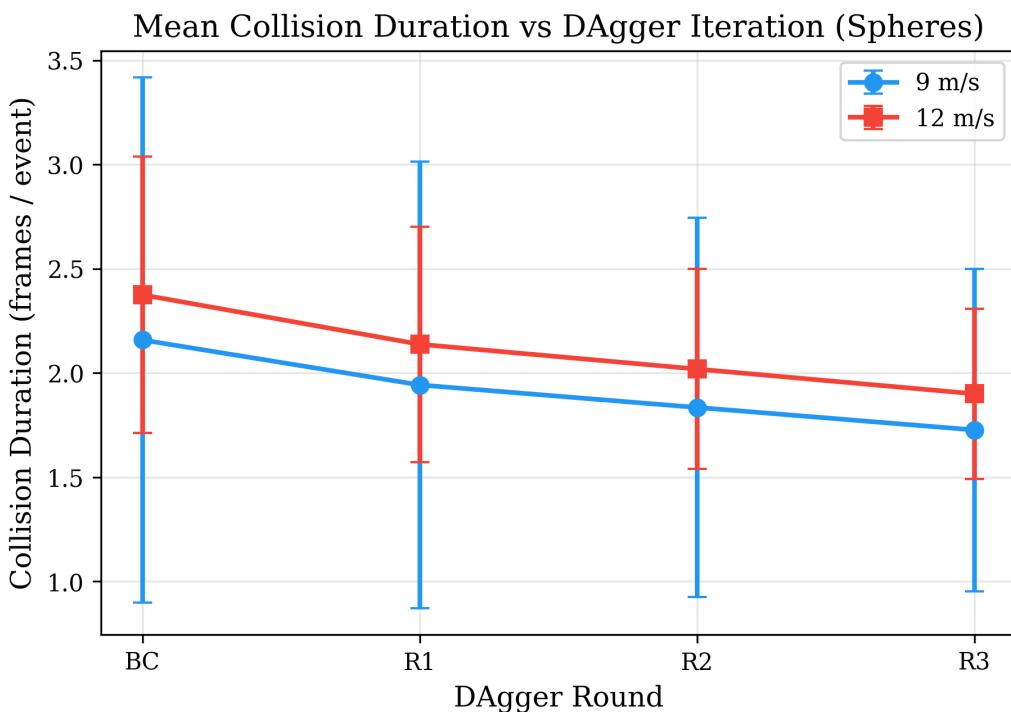


图 6.5 平均碰撞持续时间随 DAgger 轮次变化 (Spheres 环境, 9m/s 与 12m/s)。碰撞持续时间采用“含零”口径 (见第 4 章 4.8.3 节说明)。

6.9.3 行为一致性：方差收敛是 DAgger 的主要收益

图 6.6 给出了三项碰撞指标（碰撞事件次数、碰撞持续时间、碰撞率）的跨试验标准差随 DAgger 轮次的演化趋势。

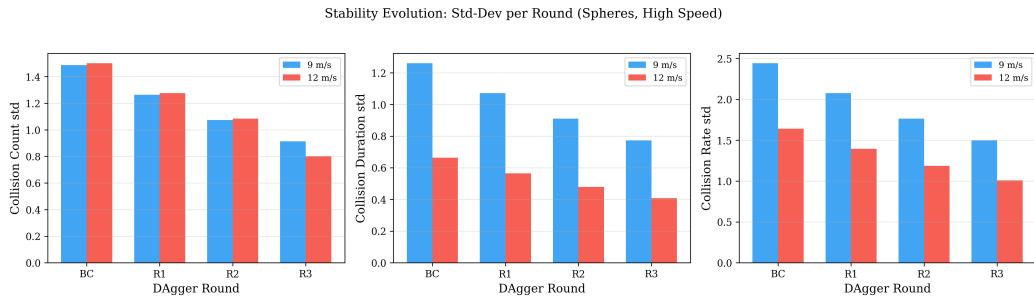


图 6.6 碰撞指标跨试验标准差随 DAgger 轮次的演化。左: Collision Count std; 中: Collision Duration std; 右: Collision Rate std。标准差下降反映了策略行为从“有时好有时差”收敛为“稳定表现”。注意: Duration std 采用含零口径 (与图 6.5 一致), 因此其趋势可能与条件分布口径 (图 6.8) 有所不同。

跨试验标准差的变化揭示了 DAgger 在本实验条件下的一个关键收益维度：**行为一致性提升**。三项指标的 std 演化分别体现以下特征：

- **Collision Count std** (左子图)：碰撞事件次数的跨试验标准差随轮次持续下降，表明策略在“发生多少次碰撞”这一维度上的行为波动性显著收敛；
- **Collision Duration std** (中子图)：碰撞持续时间的标准差总体下降但非严格单调，这与 6.8.2 节的含零口径均值趋势一致——R3 中零碰撞回合比例上升压低了含零均值，但条件口径下的持续时间存在轻微波动，影响 std 的单调性；
- **Collision Rate std** (右子图)：碰撞率的标准差随轮次收敛，表明策略在帧级碰撞频率上的表现更加稳定。

综合来看：

- BC 基线策略 (R0) 虽然在均值上已具备较好的安全性，但在 10 次独立试验间存在较大的表现波动——某些试验碰撞极少，另一些试验碰撞较多；
- DAgger 迭代通过引入当前策略诱导的闭环数据，使训练分布逐步覆盖策略实际访问的状态，减少了策略在边界状态下的不确定行为；

- 标准差下降意味着策略输出更加可预测、更稳定，这一特性在工程部署中的价值甚至超过均值改善——因为**稳定的次优策略优于不稳定的“平均最优”策略。**

该发现也解释了为何仅 18 条/轮的有限新增数据即可产生可观收益：DAgger 数据针对性地覆盖了 BC 策略的薄弱状态区域，即使数据量少，也能有效减少这些区域的行为不确定性。

6.9.4 分布可视化

为防止“仅看均值”可能掩盖的分布信息，图 6.7 和图 6.8 分别给出了碰撞事件次数与碰撞持续时间的逐轮分布。

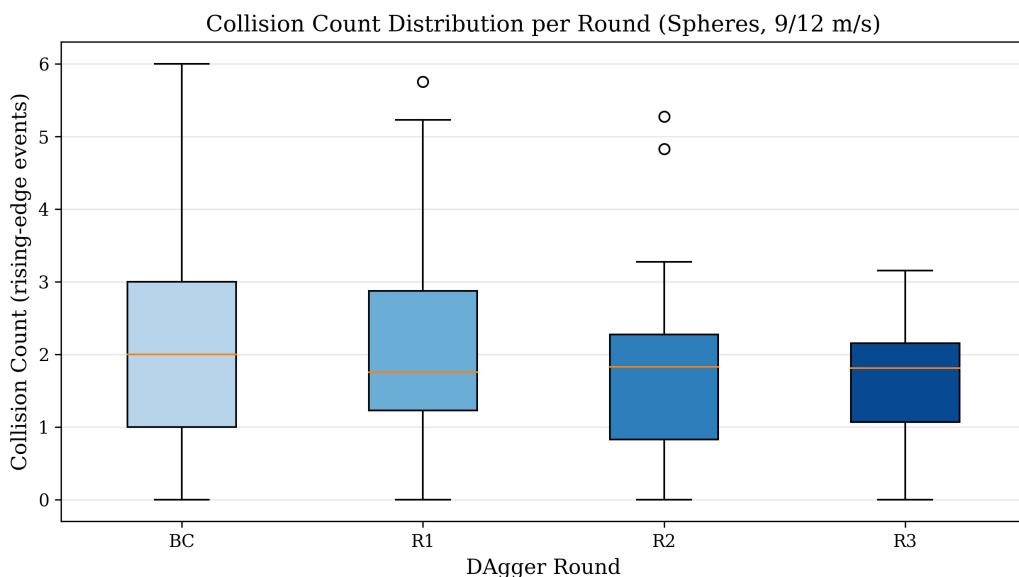


图 6.7 碰撞事件次数的逐轮分布。可以观察到分布随轮次逐步集中于低值区域。

需要明确两种口径的差异：

- **主文结论口径（含零）：**所有回合参与统计，未发生碰撞的回合记为 $\text{duration} = 0$ 。该口径用于计算跨回合均值与标准差，反映整体安全水平；
- **分布图口径（条件分布， $\text{duration} > 0$ ）：**仅统计实际发生碰撞的回合的持续时间分布。该口径用于分析“一旦碰撞发生，持续多久”，提供更细粒度的碰撞行为特征。

分布可视化为均值层面的发现提供了更细粒度的解释：碰撞事件次数的分布随轮次逐步集中于低值区域，高碰撞回合的比例下降（图 6.7）；碰撞持续时间的条件分

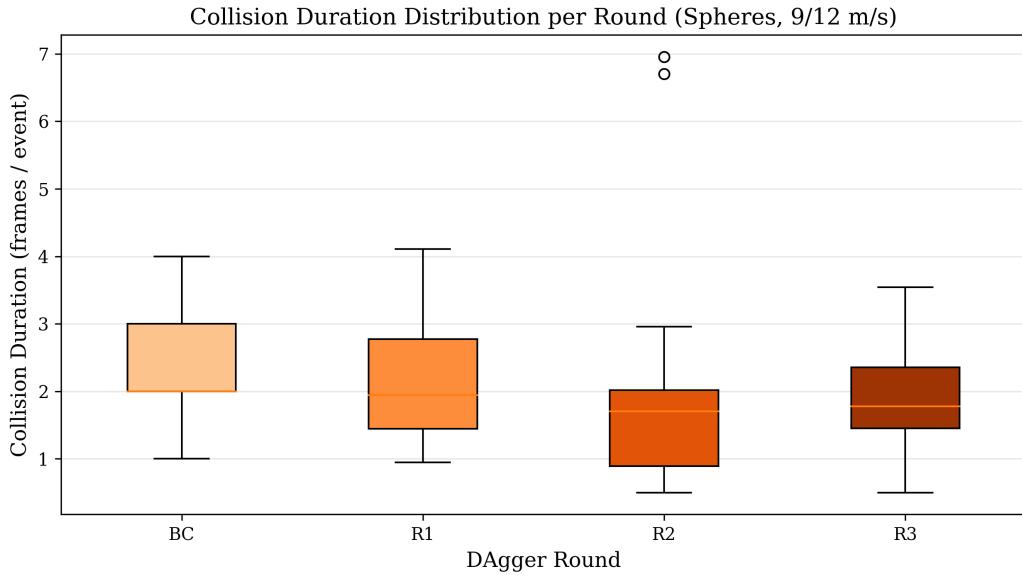


图 6.8 碰撞持续时间的逐轮分布（条件分布，仅统计 $\text{duration} > 0$ 的回合）。注意该图采用条件分布口径，主文结论采用“含零”口径，两者差异见第 4 章 4.8.3 节说明。

布在 R2 达到最紧凑状态，R3 中虽然中位数维持在较低水平，但箱体较 R2 略有展宽（图 6.8），与 6.8.2 节含零口径均值的轻微波动一致。这种非严格单调现象在有限数据增量下属于合理范围。

6.9.5 BC 与 DAgger-final 的对比：同分布与分布外性能

图 6.9 给出了 BC 基线（R0）与 DAgger 最终轮（R3）在 Spheres (ID) 与 Trees (OOD) 两类环境下各速度档位的碰撞率对比。

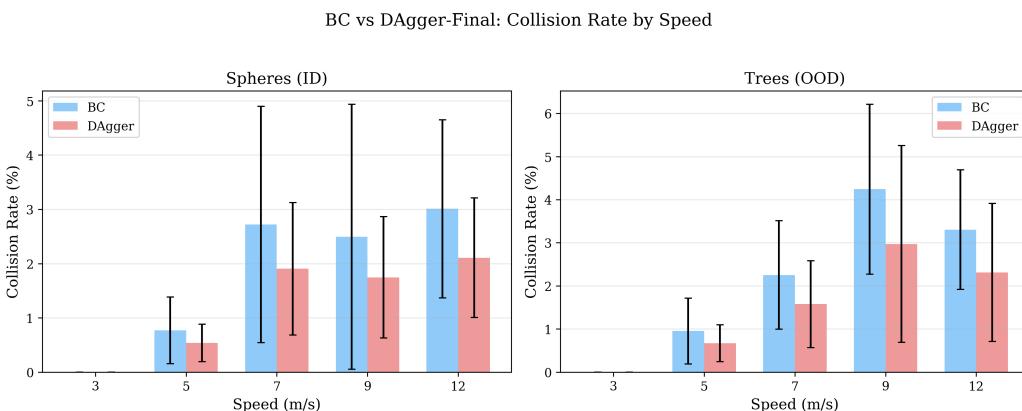


图 6.9 BC 基线与 DAgger-R3 在 Spheres (ID) 与 Trees (OOD) 环境下的碰撞率对比（各速度档）。

图 6.10 进一步给出了 Trees (OOD) 环境下的多维指标对比，包括 Command Jerk、

完成时间与平均前向速度。

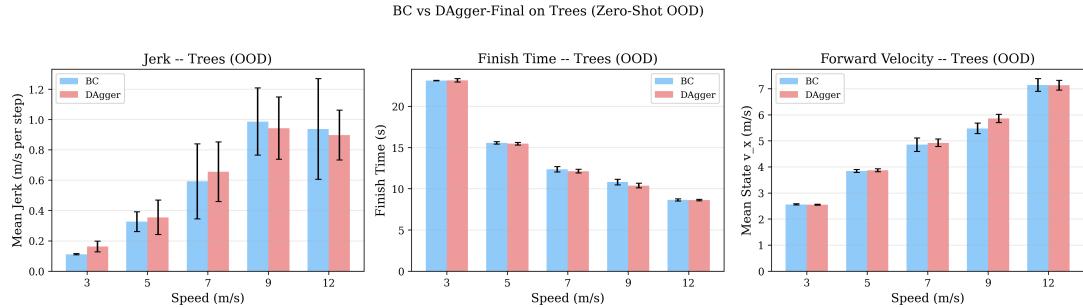


图 6.10 Trees OOD 环境下 BC 与 DAgger-R3 的多维指标对比 (Command Jerk、完成时间、平均前向速度)。

主要发现如下：

- ID 环境碰撞率进一步下降：** DAgger-R3 在 Spheres 环境的高速段碰撞率相比 BC 基线有进一步改善，与前述碰撞频次/持续时间的轮次趋势一致；
- OOD 环境零样本性能保持或改善：** 尽管 DAgger 数据仅在 Spheres 环境中采集，但在 Trees 环境的零样本评测中，DAgger-R3 并未出现性能退化，部分速度档甚至有所改善。这表明 DAgger 增强的鲁棒性具有一定的跨分布迁移能力；
- 系统效率与平滑性：** 从 Trees 环境的多维指标来看，DAgger-R3 在 Command Jerk、完成时间与平均前向速度上均保持或略有改善，说明 DAgger 数据增强未以牺牲效率或平滑性为代价。

6.9.6 DAgger 实验小结

本节的 DAgger 闭环数据增强实验在 ViT+Mamba BC 基线之上得到以下增量结论：

- 碰撞频次与持续时间进一步降低：** 3 轮 DAgger 迭代在高速段持续降低碰撞事件次数与碰撞持续时间，在强 BC 基线上实现了增量安全性提升；
- 跨试验稳定性显著增强：** 碰撞指标的标准差随轮次收敛，策略行为从“有波动”过渡为“稳定可预测”，工程部署价值显著；
- OOD 性能保持或改善：** 仅在 Spheres 环境中采集 DAgger 数据的策略，在 Trees 零样本评测中未退化，且效率与平滑性指标保持稳定。

同时需要客观指出本实验的局限性：

- 迭代轮次有限（仅 3 轮），每轮新增数据量较少（18 条轨迹），DAgger 的上界有待在更大规模采集下验证；
- 在当前环境难度下，BC 基线的成功率已趋于饱和，DAgger 的主要收益体现在碰撞细粒度指标与方差收敛而非成功率提升；
- 更困难的环境配置（更高密度障碍、更极端速度）下 DAgger 的收益幅度有待探索。

6.10 从混合架构到全 SSM 架构：MambaVision 探索实验

前述实验（6.1–6.9 节）均基于 ViT+Mamba 的混合架构：视觉编码由基于注意力机制的 ViT 完成，时序聚合由基于状态空间模型的 Mamba 完成。这一“Attention（空间）+ SSM（时间）”的混合范式在实验中取得了良好的安全性与泛化性能，构成了本文的主要贡献基础。

然而，从架构设计的理论视角审视，一个自然的问题是：**能否将空间与时间维度的建模统一到同一架构范式下？**具体而言，若将视觉编码器从 ViT 替换为同属 SSM 系列的 MambaVision[16]，形成“SSM（空间）+ SSM（时间）”的全 SSM 架构，系统性能将如何变化？

本节定位为架构统一性的前瞻探索。其价值不仅在于性能比较，更在于考察空间–时间同构建模（spatial-temporal homogeneous modeling）在端到端视觉避障任务中的可行性与潜力。即使实验结果显示全 SSM 架构的性能提升有限，该探索仍对理解 SSM 在视觉感知中的能力边界具有理论意义。

6.10.1 实验设计

MambaVision 替换实验采用严格的**控制变量设计**：

- **保持不变**：时序 Mamba 模块（4 层， $d_{model} = 192$ ）、训练流程（BC、课程学习）、评测协议（5 速度档、10 次试验、Spheres+Trees）、部署一致性机制（KeepState）；
- **唯一变量**：视觉编码器从 2-stage ViT 替换为 MambaVision。

通过该控制变量设计，性能差异可严格归因于视觉编码器的架构差异，排除其他混淆因素。

6.10.2 评测维度

替换实验将从以下四个维度进行量化评估：

1. **安全性**：碰撞率与碰撞事件次数在各速度档位下的变化；
2. **OOD 泛化**：Trees 环境中零样本测试的性能对比；
3. **推理效率**：单步推理时间与 GPU 显存占用的变化；
4. **平滑性**：Command Jerk 是否因视觉编码器变化而改变。

6.10.3 预期假设与理论动机

本实验旨在验证以下三项假设，每项假设背后均有明确的理论动机：

1. **推理效率假设**：MambaVision 利用 SSM 的线性复杂度替代 ViT 的二次复杂度自注意力，有望在同等或更优安全性下降低推理延迟与显存占用——这对嵌入式部署场景具有直接的工程价值；
2. **OOD 鲁棒性假设**：MambaVision 的混合注意力 +SSM 设计在 OOD 场景中可能提供更鲁棒的视觉表征，因为 SSM 的递归结构对空间局部特征的编码方式有别于全局注意力；
3. **架构统一性假设**：MambaVision 与时序 Mamba 形成空间--时间同构的全 SSM 架构，在工程实现上可共享算子库与优化策略，降低系统复杂度。该假设的核心价值在于架构简洁性而非单纯的性能提升。

上述假设均需通过严格实验评测确认，实验结果将在后续补充。需要强调的是：本节探索的核心贡献在于提出并验证全 SSM 架构在端到端控制中的可行性，而非追求刷新性能数字。无论结果如何，该实验均为理解 SSM 在视觉-运动控制任务中的能力边界提供了有价值的实证基础。

结论

本文面向高速端到端视觉避障控制任务，围绕“时序建模升级与训练方法、流式部署一致性、全 SSM 架构探索”三个核心维度，构建了完整的端到端闭环控制系统，并进行了系统化的实验评测与分析。以下总结本文的主要工作、核心发现与不足之处，并展望未来研究方向。

主要工作与核心发现

- (1) 提出 ViT+Mamba 高速端到端避障策略（第 3–4 章、第 6 章）。本文构建了以 ViT 空间编码、Mamba 选择性状态空间模型时序聚合与线性控制头为核心的端到端策略网络，采用行为克隆（BC）范式在 Flightmare 仿真平台中训练。在 Spheres（同分布）与 Trees（分布外）两类环境、3–12 m/s 五档速度下的系统评测表明，ViT+Mamba 相比 ViT+LSTM 基线在高速段显著降低碰撞率与碰撞事件次数，且在零样本 OOD 场景中仍保持优势。Mamba 的线性复杂度使推理延迟满足高速闭环控制的实时性要求。
- (2) 揭示流式部署中的无记忆退化现象并提出状态生命周期管理协议（第 5 章）。本文系统分析了序列模型在训练（Batch 前向）与部署（Streaming 逐步递推）之间的语义差异，揭示了错误的逐步状态重置会使序列模型退化为无记忆策略，导致碰撞率从 0% 跃升至 90%、横向漂移达 0.770 m 的严重后果。为此提出回合边界级状态生命周期管理协议，并实现运行时断言、配置锁定与可审计日志等硬防护机制，确保评测结论不受实现细节污染。
- (3) 设计部署侧动态速率限制控制平滑器 RACS（第 4 章方法，第 6 章实验）。针对 ViT+Mamba 更敏捷时序响应带来的指令抖动（Command Jerk）代价，本文在部署侧引入 RACS 模块，通过动态速率限制约束 ($\|\mathbf{v}_{\text{cmd}} - \mathbf{v}_{\text{prev}}\|_2 \leq \delta_t$) 在不修改训练过程的前提下有效降低 jerk，同时基本保持安全性优势。RACS 的计算开销低于 0.1 ms，可作为即插即用的后处理组件。
- (4) 在 BC 基线上实施 DAgger 闭环数据增强并完成系统评测（第 4、6 章）。在 ViT+Mamba BC 基线的基础上，本文进一步引入 3 轮 DAgger 迭代式闭环数据增强。实验表明，DAgger 在高速段进一步降低碰撞事件次数与碰撞持续时间，且跨试验标准差显著收敛——策略行为从“有波动”过渡为“稳定可预测”。在 Trees 零样本 OOD 评测中，DAgger 增强后的策略性能保持或略有改善，且效率与平滑性指标未退化。

不足与局限性

本文的研究仍存在以下局限：

1. **仿真环境局限**：所有实验在 Flightmare 仿真平台中完成，尚未进行真实环境（Sim-to-Real）验证。仿真与真实之间的传感噪声、气动效应与延迟特性差异可能影响策略在真实部署中的表现。
2. **静态障碍假设**：当前评测环境中的障碍物均为静态，未涉及动态障碍（如运动物体、其他飞行器）的避障场景。
3. **单一传感模态**：策略仅使用单目深度图像作为视觉输入，未融合 RGB 图像、激光雷达或事件相机等多模态信息。
4. **DAgger 规模有限**：本文实施的 DAgger 仅执行了 3 轮迭代，每轮新增 18 条轨迹，数据增量较小。DAgger 的安全性提升上界有待在更大采集规模、更多迭代轮次与更高难度环境配置下进一步验证。
5. **RACS 的启发式特性**：RACS 的动态速率上界 δ_t 基于最小深度的启发式规则，缺乏形式化安全保证。

未来工作展望

基于本文的研究基础，后续工作将在以下方向展开：

1. **DAgger 更大规模扩展**：在当前 3 轮 DAgger 的基础上，增加迭代轮次（5 轮以上）、引入更困难的环境配置（更高密度障碍、更极端速度）、扩大每轮采集规模，系统探索 DAgger 在高速避障任务中的收益上界。
2. **MambaVision 视觉编码器替换**：在保持时序 Mamba 与部署一致性机制不变的条件下，将空间编码器替换为 MambaVision，量化空间--时间统一 SSM 架构在安全性、泛化能力与推理效率方面的收益与边界。
3. **仿真到真实迁移**：结合域随机化、系统辨识与真实飞行数据校正，将策略部署到真实四旋翼平台上进行验证。
4. **动态障碍应对**：扩展评测环境以包含动态障碍物，探索策略在时变环境中的避障能力。

5. **多模态融合**: 探索将 RGB 图像、事件相机等多模态信息融合到视觉编码器中, 提升感知鲁棒性。
6. **RACS 向风险自适应升级**: 将 RACS 的启发式规则升级为基于 TTC 或碰撞概率的数据驱动风险估计, 或与 CBF/MPSC 等安全证书方法进行融合, 提供更严格的安全保证。

参考文献

- [1] Loquercio A, Kaufmann E, Ranftl R, et al. Learning high-speed flight in the wild[J]. Science Robotics, 2021, 6(59).
- [2] Mur-Artal R, Tardós J D. Orb-slam2: An open-source slam system for monocular, stereo, and rgbd cameras[J]. IEEE Transactions on Robotics, 2017, 33(5): 1255–1262.
- [3] Qin T, Li P, Shen S. Vins-mono: A robust and versatile monocular visual-inertial state estimator[J]. IEEE Transactions on Robotics, 2018, 34(4): 1004–1020.
- [4] Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning[C]. The International Journal of Robotics Research. [S.l.: s.n.], 2011.
- [5] Faust A, Palunko I, Cruz P, et al. Faster: Fast and safe trajectory planner for flights in unknown environments[J]. IEEE Transactions on Robotics, 2018.
- [6] Pomerleau D A. ALVINN: An autonomous land vehicle in a neural network[J]. Advances in Neural Information Processing Systems (NeurIPS), 1989.
- [7] Bojarski M, Del Testa D, Dworakowski D, et al. End to end learning for self-driving cars[C]. arXiv preprint arXiv:1604.07316. [S.l.: s.n.], 2016.
- [8] Loquercio A, Maqueda A I, Del Blanco C R, et al. Dronet: Learning to fly by driving[C]. IEEE Robotics and Automation Letters: volume 3. 2018: 1088–1095.
- [9] Sadeghi F, Levine S. CAD2RL: Real single-image flight without a single real image[C]. Robotics: Science and Systems (RSS). [S.l.: s.n.], 2017.
- [10] Gandhi D, Pinto L, Gupta A. Learning to fly by crashing[J]. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017.
- [11] Kaufmann E, Loquercio A, Ranftl R, et al. Deep drone racing: Learning agile flight in dynamic environments[C]. Proceedings of the Conference on Robot Learning (CoRL). [S.l.: s.n.], 2018.
- [12] Kaufmann E, Bauersfeld L, Loquercio A, et al. Champion-level drone racing using deep reinforcement learning[J]. Nature, 2023.
- [13] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735–1780.
- [14] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in Neural Information Processing Systems (NeurIPS), 2017.
- [15] Gu A, Dao T. Mamba: Linear-time sequence modeling with selective state spaces[J]. arXiv preprint

- arXiv:2312.00752, 2023.
- [16] Hatamizadeh A, Kautz J. Mambavision: A hybrid mamba-transformer vision backbone[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2025.
- [17] Brunke L, Greeff M, Hall A, et al. Safe learning in robotics: From learning-based control to safe reinforcement learning[J]. Annual Review of Control, Robotics, and Autonomous Systems, 2022, 5: 411–444.
- [18] Cheng R, Orosz G, Murray R M, et al. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks[J]. arXiv preprint arXiv:1903.08792, 2019.
- [19] Wabersich K, Zeilinger M N. A predictive safety filter for learning-based control of constrained nonlinear dynamical systems[J]. arXiv preprint arXiv:1803.08552, 2018.
- [20] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[J]. arXiv preprint arXiv:2010.11929, 2020.
- [21] Ross S, Gordon G, Bagnell D. A reduction of imitation learning and structured prediction to no-regret online learning[C]. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS). [S.l.: s.n.], 2011.
- [22] Song Y, Sun J, Liu S, et al. Flightmare: A flexible quadrotor simulator[C]. Proceedings of the Conference on Robot Learning (CoRL). [S.l.: s.n.], 2021.
- [23] 高翔, 张涛, 刘毅, 等. 视觉 SLAM 研究进展[J]. 机器人, 2019, 41(6): 768–776.
- [24] 张弓, 方正, 陈思哲. 移动机器人视觉惯性 SLAM 研究进展[J]. 自动化学报, 2018, 44(5): 769–786.
- [25] 王猛, 赵清杰, 翟鹏. 无人机三维路径规划与避障方法研究[J]. 信息与控制, 2020, 49(3): 328–336.
- [26] 李志刚, 刘超, 王志超. 基于改进 RRT 算法的多旋翼无人机避障航迹规划[J]. 中国农业工程学报, 2019, 35(6): 59–67.
- [27] Foehn P, Kaufmann E, Romero A, et al. Agilicious: Open-source and open-hardware agile quadrotor for vision-based flight[J]. Science Robotics, 2022, 7(67).
- [28] Kaufmann E, Loquercio A, Ranftl R, et al. Deep drone acrobatics[C]. Robotics: Science and Systems (RSS). [S.l.: s.n.], 2020.
- [29] Xing Z, Romero A, Scaramuzza D. Vision transformers for end-to-end vision-based quadrotor obstacle avoidance[J]. arXiv preprint arXiv:2405.10391, 2024.
- [30] Gu A, Goel K, Re C. Efficiently modeling long sequences with structured state spaces[J]. Interna-

- tional Conference on Learning Representations (ICLR), 2022.
- [31] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278–2324.
- [32] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[J]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016: 770–778.
- [33] Zhou B, Gao F, Wang L, et al. Robust and efficient quadrotor trajectory generation for fast autonomous flight[J]. IEEE Robotics and Automation Letters, 2019, 4(4): 3529–3536.
- [34] Zhou X, Wang Z, Ye H, et al. EGO-Planner: An ESDF-free gradient-based local planning framework for quadrotors[J]. IEEE Robotics and Automation Letters, 2021, 6(2): 478–485.
- [35] 何承坤, 戴明, 张瑞. 四旋翼无人机轨迹优化与跟踪控制研究综述[J]. 控制与决策, 2021, 36(7): 1537–1556.
- [36] 雷志勇, 陈世超, 刘尚旺. 基于深度强化学习的无人机视觉避障方法[J]. 自动化学报, 2020, 46(4): 796–805.
- [37] 陈杰, 李擎, 王莉. 深度强化学习在无人机自主导航中的研究进展[J]. 航空学报, 2023, 44(4): 1–18.

附录 A ***

附录相关内容…

附录 B Maxwell Equations

因为在柱坐标系下， $\bar{\mu}$ 是对角的，所以 Maxwell 方程组中电场 \mathbf{E} 的旋度
所以 \mathbf{H} 的各个分量可以写为：

$$H_r = \frac{1}{i\omega\mu_r} \frac{1}{r} \frac{\partial E_z}{\partial\theta} \quad (B-1a)$$

$$H_\theta = -\frac{1}{i\omega\mu_\theta} \frac{\partial E_z}{\partial r} \quad (B-1b)$$

同样地，在柱坐标系下， $\bar{\epsilon}$ 是对角的，所以 Maxwell 方程组中磁场 \mathbf{H} 的旋度

$$\nabla \times \mathbf{H} = -i\omega \mathbf{D} \quad (B-2a)$$

$$\left[\frac{1}{r} \frac{\partial}{\partial r} (r H_\theta) - \frac{1}{r} \frac{\partial H_r}{\partial\theta} \right] \hat{\mathbf{z}} = -i\omega \bar{\epsilon} \mathbf{E} = -i\omega \epsilon_z E_z \hat{\mathbf{z}} \quad (B-2b)$$

$$\frac{1}{r} \frac{\partial}{\partial r} (r H_\theta) - \frac{1}{r} \frac{\partial H_r}{\partial\theta} = -i\omega \epsilon_z E_z \quad (B-2c)$$

由此我们可以得到关于 E_z 的波函数方程：

$$\frac{1}{\mu_\theta \epsilon_z} \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial E_z}{\partial r} \right) + \frac{1}{\mu_r \epsilon_z} \frac{1}{r^2} \frac{\partial^2 E_z}{\partial\theta^2} + \omega^2 E_z = 0 \quad (B-3)$$

攻读学位期间发表论文与研究成果清单

- [1] 高凌. 交联型与线形水性聚氨酯的形状记忆性能比较 [J]. 化工进展, 2006, 532 - 535. (核心期刊) .

致谢

感谢人民，感谢党。

本论文的工作是在导师……。

作者简介

本人…。