NavRL：在动态环境中学习安全飞行

Zhefan Xu, Xinming Han, Haoyu Shen, Hanyu Jin, and Kenji Shimada



图 1. 一架定制四旋翼无人机使用本文提出的 NavRL 框架在动态环境中导航。机器人实现了安全导航，并能有效避开静态与动态障碍物。

摘要—在动态环境中安全飞行，要求无人机（Unmanned Aerial Vehicle, UAV）在穿越布满移动障碍物的杂乱空间时做出有效决策。传统方法通常将决策分解为预测与规划等层级模块。这类人工设计的系统虽然在特定场景下表现良好，但环境条件变化时可能失效，且往往需要细致调参。此外，为追求计算效率而采用的不准确数学模型假设与简化，也可能导致其解是次优的。为克服这些局限，本文提出 NavRL 框架——一种基于近端策略优化（Proximal Policy Optimization, PPO）算法的深度强化学习导航方法。NavRL 采用精心设计的状态与动作表示，使学习到的策略能够在静态与动态障碍物并存的情况下做出安全决策，并实现从仿真到真实飞行的零样本迁移（Zero-shot Transfer）。此外，受速度障碍（Velocity Obstacle）概念启发，本方法为训练好的策略引入一种简单而有效的安全护盾（Safety Shield），以缓解神经网络黑盒特性带来的潜在失效。为加速收敛，训练流程基于 NVIDIA Isaac Sim 实现，可同时并行训练数千架四旋翼。仿真与实物实验[1]表明，本方法能够保证动态环境中的安全导航，且与基准方法相比碰撞次数最少。

关键词—空中系统：感知与自主，强化学习，避碰

I. 引言

主无人机（UAV）被广泛应用于探索 [1]、搜索救援 [2] 自和巡检 [3] 等任务。这些任务常发生在动态环境中，需要有效的避碰能力。传统方法依赖人工设计的算法与层级模块，导致系统过于复杂、参数难以调节。相比之下，强化学习（Reinforcement Learning, RL）让无人机通过经验学习决策，具有更好的适应性和更优的性能。因此，开发基于 RL 的导航方法对提升无人机在动态环境中的安全性至关重要。

开发适用于真实部署的 RL 导航方法面临若干挑战。第一，强化学习需要让机器人通过碰撞经验进行训练，因此学习过程必须在仿真环境中进行。仿真与真实传感信息之间（尤其是相机图像）存在差距，由此产生仿真到现实迁移（Sim-to-Real Transfer）问题。已有工作尝试缩小仿真与真实环境之间的差距 [4][5][6]，或直接在真实环境中训练机器人 [7][8][9]。然而，这些方法通常需要额外的训练步骤，在真实场景中训练时数据效率也较低。第二，即便训练出的 RL 策略表现令人满意，由于神经网络的黑盒特性，保证安全仍然困难，需要有效的安全机制防止严重失效 [10][11] [12][13]。最后，训练 RL 策略需要大量机器人经验，部分已有方法 [14][15][16] 仅用单个机器人采集数据，数据多样性有限、并行探索机会少，收敛速度往往较慢。

为解决上述问题，本文基于近端策略优化（PPO）算法 [17] 提出名为 NavRL 的深度强化学习导航框架。该框架在感知模块之上构建状态与动作表示，专门针对动态环境中的避碰设计，具备零样本仿真到现实迁移能力。为防止严重失效，本文利用速度障碍（Velocity Obstacle, VO）[18] 概念，通过线性规划优化 RL 策略网络的输出动作，构建简单而有效的安全护盾。为加快训练收敛，本文设计了并行训练流程，可在 NVIDIA Isaac Sim 中同时仿真数千架四旋翼。通过在多种环境中开展大量仿真与实物飞行实验，验证了该框架能够保证安全导航。图 1 展示了无人机使用 NavRL 框架在动态环境中导航的示例。本文的主要贡献如下：

- **NavRL 导航框架：**本文提出一种基于强化学习的无人机导航系统，保证动态环境中的安全自主飞行。NavRL 导航框架已在 GitHub 开源[2]。
- **策略动作安全护盾：**基于速度障碍概念，在策略网络的输出动作上施加安全护盾以提升安全性。
- **实物飞行实验：**在多种真实环境中开展实验，验证了本方法的安全导航能力与零样本仿真到现实迁移的有效性。

II. 相关工作

动态环境中的无人机导航研究多依赖基于规则、人工设计的算法 [19][20][21]，这类方法较为复杂，且条件变化时需要细致调参。相比之下，基于学习的方法降低了系统复杂度，对多变环境适应性更好。本节主要将基于学习的导航方法分为监督学习方法和强化学习方法两类，同时也承认存在其他方法。

基于监督学习的方法：此类方法使用带标注的数据集训练网络。早期方法 [8][22] 将机器人部署到真实环境中采集图像，并人工标注真值动作。类似地，一些方法 [7][23] 不直接输出决策，而是预测输入图像的安全性，再用人工设计的算法控制机器人。

上述方法需要人工标注的真实数据，且常因数据不足导致泛化能力有限。Loquercio 等人 [24] 利用自动驾驶数据集训练网络实现导航，受益于其庞大的数据量。Jung 等人 [25] 使用基于学习的检测器使无人机在竞速中穿越门框，[26] 则在此场景中应用迭代学习控制以降低跟踪误差。也有工作利用多种机器人的真实经验构建视觉导航基础模型 [27][28][29]。Simon 等人 [30] 借助深度估计方法 [31]，展示了搭载单目相机的无人机的避碰能力。[32] 提出一种

稿件收到日期：2024 年 9 月 23 日；修改日期：2025 年 1 月 3 日；录用日期：2025 年 2 月 11 日。
本文经副编辑与审稿人评审后，由编辑 Aleksandra Faust 推荐发表。
Zhefan Xu、Xinming Han、Haoyu Shen、Hanyu Jin 和 Kenji Shimada 就职于卡内基梅隆大学机械工程系，地址：5000 Forbes Ave, Pittsburgh, PA, 15213, USA。zhefanx@andrew.cmu.edu
数字对象标识符（DOI）：见本页顶部。
[1]实验视频链接：https://youtu.be/EbeJW8-YlvI

[2]软件地址：https://github.com/Zhefan-Xu/NavRL

基于语义图像的命令式学习（Imperative Learning）方法，实现语义感知的局部导航。

基于强化学习的方法：与监督学习方法相比，强化学习方法受益于仿真中丰富的数据。一些采用 Q 学习（Q-learning）[14][15][33] 或价值学习（Value Learning）[34][35] 的方法已实现成功导航。然而，这些方法局限于离散动作空间，可能导致性能次优。

采用演员-评论家（Actor-Critic）结构的策略梯度（Policy Gradient）方法 [16][36] 能够在连续动作空间中控制机器人，因而日益流行。Kaufmann 等人 [37] 在仿真中训练策略，在无人机竞速中击败了人类冠军，其他研究则进一步提升了控制性能 [38][39]。Song 等人 [6] 通过从拥有特权信息的教师策略中蒸馏知识，构建了基于视觉的网络；Xing 等人 [40] 则应用对比学习改进图像编码。[41] 提出一套面向无人机相关任务的完整强化学习训练流程。

为保证安全，[42] 引入基于可达-规避网络（Reach-Avoid Network）的恢复策略以防止失效。类似地，Kochdumper 等人 [13] 利用可达性分析将不安全的策略动作投影到安全区域。然而，该方法需要预先计算可达集，计算代价随动作维度呈指数增长。

现有方法大多面向静态环境中的导航与避碰。部分方法（如 [36]）展示了动态障碍物规避能力，但未处理复杂静态场景中的避碰。[42] 的方法能同时处理静态与动态障碍物，但将动态障碍物视为静态，可能导致性能次优。近期以安全为核心的方法要么计算代价高，要么需要大量训练。这些挑战加上仿真到现实迁移的困难，促使本文提出一个同时规避静态与动态障碍物、保证安全导航的框架。

III. 方法

NavRL 导航框架如图 2 所示。障碍物感知系统处理 RGB-D 图像与机器人状态，生成静态与动态障碍物的表示（第 III-A 节）。第 III-B 节详细说明如何将这些障碍物表示转换为网络输入状态，并给出机器人动作与训练奖励的定义。训练阶段采用 PPO 算法与演员-评论家网络结构训练策略网络与价值网络，见第 III-C 节。部署阶段对 RL 策略网络的输出动作施加安全护盾以保证安全（第 III-D 节）。

A. 障碍物感知系统

由于静态与动态障碍物性质不同，感知系统对二者分别处理，如图 2 所示。静态障碍物形状与尺寸任意，采用离散化格式（如占据体素地图）表示更为准确；而动态障碍物通常建模为刚体，用带有估计速度信息的包围盒表示。二者采用不同表示的原因在于其特性不同：静态障碍物有时几何形状不规则，离散表示更适合精确描述形状；动态障碍物通常是刚体，需要将其状态作为整体实体刻画。此外，用动态障碍物更新占据地图会引入噪声与延迟，这进一步说明需要分开表示。对于静态障碍物感知，本文构建固定内存大小的三维占据体素地图，大小由适合该环境的最大体素数决定，每个体素的占据数据存储在预分配数组中。该设计使占据信息的访问时间复杂度为常数（$\mathcal{O}(1)$）。每个时间步，根据最新深度图像递归更新每个体素的占据对数概率，并遍历检测到的动态障碍物包围盒，清除其对应的占据数据。值得一提的是，该静态占据地图可在线实时生成，无需任何预建数据。

由于轻量级无人机相机的深度图像噪声大、机载计算机算力有限，三维动态障碍物检测颇具挑战。为在最小计算开销下实现准确检测，本文提出一种集成方法，结合基于 [43] 构建的两个轻量级检测器。第一个是 U-depth 检测器 [44]，它将原始深度图像转换为 U-depth 图（类似俯视图），并使用连续线段分组算法检测障碍物的三维包围盒。第二个称为 DBSCAN 检测器，对深度图像生成的点云应用 DBSCAN 聚类算法，通过分析每个聚类内的边界点确定障碍物中心与尺寸。然而，由于输入数据噪声大，两个检测器都可能产生大量误检。本文的集成方法通过寻找两个检测器的相互一致结果来解决这一问题。此外，为区分静态与动态障碍物，将三维检测结果重投影到图像平面得到二维包围盒，再用轻量级 YOLO 检测器对其分类以识别动态障碍物。

动态障碍物速度由跟踪模块估计，分为数据关联与状态估计两个阶段。数据关联阶段的目标是建立相邻时间步中检测到的障碍物之间的对应关系。为减少错误匹配，为每个障碍物构建包含位置、包围盒尺寸、点云大小和点云标准差的特征向量，再根据候选匹配障碍物之间的相似度分数确定匹配。状态估计阶段使用卡尔曼滤波器估计障碍物速度，并采用恒加速度模型刻画速度随时间的变化。需要指出，虽然本框架的感知系统基于相机，但只要能提供类似的障碍物表示，其他传感器同样适用。感知系统中的动态障碍物检测与跟踪基于 [43] 的方法，读者可在其中找到更多细节与实现说明。

B. 强化学习建模

导航任务被建模为由元组 (S, A, P, R, γ) 定义的马尔可夫决策过程（Markov Decision Process, MDP），其中 S 为状态空间（机器人内部数据与传感数据），A 为动作空间，转移函数 $P(s_{t+1}|s_t, a_t)$ 刻画环境动力学，奖励函数 $R(s_t, a_t)$ 鼓励到达目标的行为并惩罚碰撞与低效动作。目标是学习最大化期望累积奖励的最优策略 $\pi^*(a_t|s_t)$：

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t)\right], \tag{1}$$

其中 $\gamma \in [0, 1]$ 为未来奖励的折扣因子。该建模使得可以用强化学习训练机器人在动态环境中安全导航。

状态：作为策略的输入，状态必须包含与导航和避碰相关的全部信息。在图 2 所示的系统框架中，状态由三部分组成：机器人内部状态、动态障碍物和静态障碍物。机器人内部状态提供机器人到导航目标的方向与距离以及当前速度，定义为：

$$S_{int} = [\frac{P_g^G - P_r^G}{\|P_g^G - P_r^G\|}, \|P_g^G - P_r^G\|, V_r^G]^T, \tag{2}$$

其中 P_r 与 P_g 分别表示机器人位置与目标位置，V_r 为机器人当前速度。上标 $(\cdot)^G$ 表示向量在“目标坐标系”中表达，该坐标系原点位于机器人起始位置，x 轴与从起始位置 P_s 指向目标位置的向量对齐，y 轴平行于地平面。这种目标坐标变换降低了对全局坐标系的依赖，提升了 RL 训练的整体收敛速度，也同样用于障碍物状态表示的定义。

动态障碍物用一个二维矩阵表示：

$$S_{dyn} = [\mathcal{D}_1, ..., \mathcal{D}_{N_d}]^T,\ S_{dyn} \in \mathbb{R}^{N_d \times M},\ \mathcal{D}_i \in \mathbb{R}^M. \tag{3}$$

其中 $\mathcal{D}_i$ 表示距离机器人第 i 近的动态障碍物的状态向量，表达为：

$$\mathcal{D}_i = [\frac{P_{o_i}^G - P_r^G}{\|P_{o_i}^G - P_r^G\|}, \|P_{o_i}^G - P_r^G\|, V_{o_i}^G, \dim(o_i)]^T, \tag{4}$$

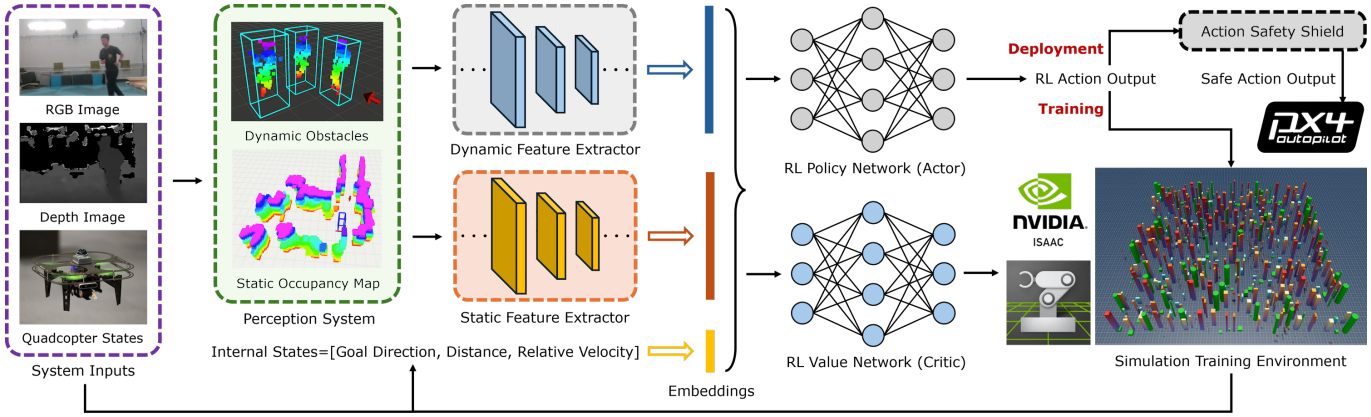
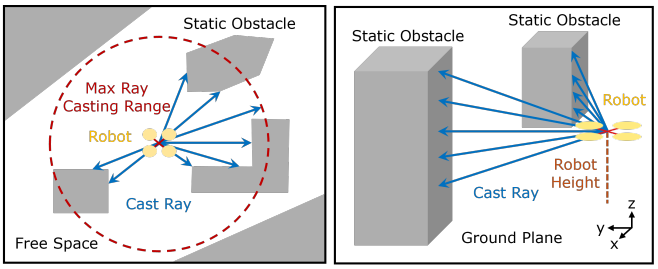


图 2. NavRL 框架。感知系统处理 RGB-D 图像与机器人内部状态，生成静态与动态障碍物的表示。这些表示被送入两个特征提取器，产生的状态嵌入与机器人内部状态拼接。训练阶段采用演员-评论家网络结构，在 NVIDIA Isaac Sim 环境中并行训练机器人。部署阶段，策略网络生成的动作经安全护盾机制进一步修正，以保证机器人控制安全。



(a) Top-down View of Map Ray Casting (b) 3D Side View of Map Ray Casting

图 3. 地图光线投射示意图，仅显示最大范围内的光线。(a) 360 度水平投射光线的俯视图。(b) 竖直平面内光线的侧视图。

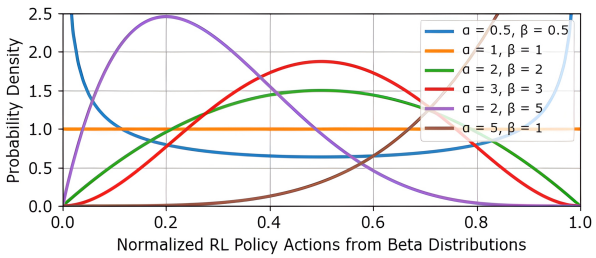


图 4. 基于 Beta 分布的 RL 策略动作示例可视化。

其中 P_{o_i} 与 V_{o_i} 分别表示动态障碍物的中心位置与速度，$\dim(o_i)$ 表示障碍物的高度与宽度。动态障碍物数量 N_d 预先设定，若实际检测到的障碍物少于该上限，剩余状态向量置零。内部状态与障碍物状态中的相对位置向量都被拆分为单位向量及其模长，实验中该做法的收敛速度略快。

与动态障碍物不同，静态障碍物以地图体素表示，无法直接输入神经网络。因此，本文从机器人位置出发对地图进行三维光线投射（Ray Casting）。如图 3a 所示，在最大范围内按用户设定的投射角度间隔向水平各方向投射光线；图 3b 展示了竖直平面内的相同操作。对竖直平面内的每个斜向投射角 θ_i，将水平平面内所有光线的长度记录为向量 R_{θ_i}。超出最大范围的光线长度设为最大范围加一个小偏移量，以便识别无障碍物的情况。将所有竖直斜向角对应的光线长度向量堆叠，即得到静态障碍物表示：

$$S_{stat} = [R_{\theta_0}, ..., R_{\theta_{N_v}}], S_{stat} \in \mathbb{R}^{N_h \times N_v}, R_{\theta_i} \in \mathbb{R}^{N_h}, \quad (5)$$

其中 N_v 与 N_h 分别表示竖直与水平平面内的光线数量，由投射角度间隔与竖直视场角决定。与使用图像数据作为状态表示不同，本文的 RL 状态表示在仿真与现实之间差异极小，有利于仿真到现实迁移。

动作：每个时间步向机器人提供速度控制量 $V_{ctrl} \in \mathbb{R}^3$ 以进行导航与避碰。选择速度控制的原因在于，较高层级的控制在不同平台间具有更好的可迁移性与泛化性，便于仿真到现实迁移。此外，与低层控制相比，速度指令更具可解释性，也更便于人类监督。RL 策略不直接输出速度值，而是先推断归一化速度 $\hat{V}^G_{ctrl}$，最终输出速度表示为：

$$V^G_{ctrl} = v_{lim} \cdot (2 \cdot \hat{V}^G_{ctrl} - 1), \ \hat{V}^G_{ctrl} \in [0, 1], \quad (6)$$

其中 v_{lim} 为用户设定的最大速度。最终速度在目标坐标系中表达（同状态建模所述），施加到机器人前需进行坐标变换。其他建模方式中 RL 策略必须自行学习动作上下限，且训练后难以调整动作范围；相比之下本方法更为灵活。为将网络输出约束在 $[0, 1]$ 范围内，模型输出 Beta 分布的参数 (α, β)，如图 4 所示。已有研究证明，在动作空间受限时，基于 Beta 分布的策略无偏，且比基于高斯分布的策略收敛更快 [45]。训练过程中，利用生成的参数从 Beta 分布中采样以鼓励探索；部署时，使用 Beta 分布的均值作为归一化速度输出。

奖励：RL 奖励函数在每个时间步计算，由多个分量组成：

$$r = \lambda_1 r_{vel} + \lambda_2 r_{ss} + \lambda_3 r_{ds} + \lambda_4 r_{smooth} + \lambda_5 r_{height}, \quad (7)$$

其中 $r_{(\cdot)}$ 表示某一类奖励，权重为 λ_i。下文逐一说明各项奖励。

a) 速度奖励 r_{vel}：速度奖励鼓励机器人采用朝向目标位置的速度：

$$r_{vel} = \frac{P_g - P_r}{\|P_g - P_r\|} \cdot V_r, \ \text{where } P_g, P_s, V_r \in \mathbb{R}^3. \quad (8)$$

该式奖励与位置到目标方向更一致且速度更高的运动。

b) 静态安全奖励 r_{ss}：静态安全奖励保证机器人与静态障碍物保持安全距离。根据式 5 中的静态障碍物状态，定义为：

$$r_{ss} = \frac{1}{N_h N_v} \sum_{i=1}^{N_h} \sum_{j=1}^{N_v} \log S_{stat}(i, j). \quad (9)$$

该式利用光线距离计算到静态障碍物的平均对数距离。机器人与障碍物距离越大，奖励越高。

c) 动态安全奖励 r_{ds}：与静态安全奖励类似，动态安全奖励鼓励机器人规避动态障碍物，表达为：

$$r_{ds} = \frac{1}{N_d}\sum_{i=1}^{N_d} \log\|P_r - P_{o_i}\|. \tag{10}$$

d) 平滑奖励 r_{smooth}：平滑奖励惩罚控制输出的突变，写为：

$$r_{smooth} = -\|V_r(t_i) - V_r(t_{i-1})\|, \tag{11}$$

即计算当前与上一时间步机器人速度之差的 L2 范数。

e) 高度奖励 r_{height}：高度奖励用于防止机器人通过飞得过高来规避障碍物，可写为：

$$r_{height} = -(min(|P_{r,z} - P_{s,z}|, |P_{r,z} - P_{g,z}|))^2, \tag{12}$$

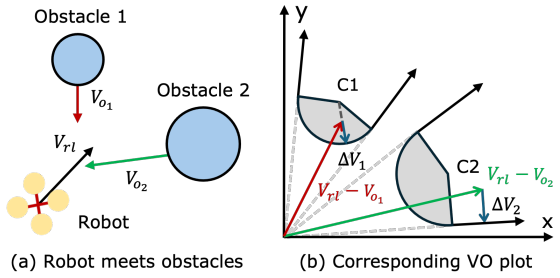
仅当当前高度 $P_{r,z}$ 超出起始高度 $P_{s,z}$ 与目标高度 $P_{g,z}$ 所界定的范围时生效。

C. 网络设计与策略训练

状态表示由多个部分组成，因此在输入 RL 策略网络前需要预处理。静态与动态障碍物都表示为二维矩阵，因此使用卷积神经网络（Convolutional Neural Network, CNN）提取特征，并将其转换为一维特征嵌入，如图 2 所示。这些嵌入与机器人内部状态拼接，构成策略网络与价值网络的完整输入特征。本文采用近端策略优化（PPO）算法 [17] 训练演员（策略）网络与评论家（价值）网络，二者均由多层感知机实现。训练在 NVIDIA Isaac Sim 中进行，通过同时采集数千架四旋翼的数据实现并行训练。训练环境为包含静态与动态障碍物的类森林场景。每个机器人在随机位置生成并分配随机目标，发生碰撞或回合结束时重置。为提升学习效率，本文采用课程学习（Curriculum Learning）策略：环境初始障碍物密度较低，当成功率超过指定阈值后逐步提高密度。实验表明，该策略使 RL 策略在复杂环境中取得更高的导航成功率。

D. 策略动作安全护盾

由于神经网络的黑盒特性，本文设计了安全护盾机制，防止训练所得策略导致严重失效。策略输出速度指令控制机器人，因此本文引入速度障碍（VO）[18] 概念评估策略动作的安全性，并在可能导致碰撞时进行修正。速度障碍表示在给定未来时间范围内会导致碰撞的机器人速度集合。对于图 5a 中机器人遭遇障碍物的场景，相应的速度障碍区域在图 5b 中可视化。该示例中有两个以不同速度运动的障碍物，每个障碍物根据其相对机器人的位置与速度对应一个速度障碍区域。每个速度障碍区域由两部分组成：以 C1 或 C2 为圆心、半径等于机器人与障碍物尺寸之和再加用户设定安全距离的圆形区域，以及从原点延伸的锥形区域（不含虚线部分）。机器人相对障碍物 1 和障碍物 2 的速度（红色与绿色箭头）位于速度障碍区域内，表明未来将发生碰撞。受 [46] 启发，本文采用类似方法计算离开速度障碍区域所需的最小速度变化（ΔV_1 与 ΔV_2）。网络产生的策略动作定义为速度 V_{rl}。若该动作不在任何速度障碍区域内，则令安全动作 V_{safe} 等于策略动作；否则，构建如下优化问题，将策略动作投影到安全区域：



(a) Robot meets obstacles (b) Corresponding VO plot

图 5. 基于速度障碍方法确定安全速度区域的示意图。(a) 机器人遭遇两个障碍物的示例场景。(b) 对应的速度障碍图，蓝色箭头表示离开 VO 区域所需的最小速度变化。

$$\min_{V_{safe} \in \mathbb{R}^3} \quad \left\|V_{safe} - V_{rl}\right\|, \tag{13a}$$

$$\text{s.t.} \quad (V_{safe} - (V_{rl} - V_{o_i} + \Delta V_i)) \cdot \Delta V_i \geq 0, \tag{13b}$$

$$V_{min} \leq V_{safe} \leq V_{max}, \tag{13c}$$

$$\forall i \in \{1, \ldots, N\}, \tag{13d}$$

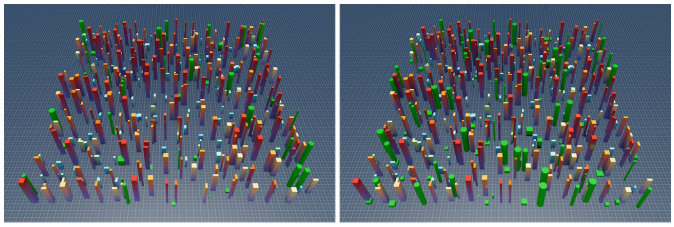
其中式 13b 根据所需速度变化定义超平面，保证安全动作位于速度障碍区域之外；式 13c 对安全动作施加控制上下限。将安全动作约束在超平面一侧，在障碍物较多时可能过于保守。但由于策略动作只是偶尔失效，这种保守性对整体性能影响不大，且在大多数场景中有助于保证安全。对于静态障碍物，利用每条投射光线确定障碍物的中心位置与半径，并将其速度设为零；对于动态障碍物，用一个或多个球体将其包围。

IV. 结果与讨论

为评估本文框架，本节给出不同配置下的训练结果，并在多种环境中开展仿真与实物飞行测试。策略在 NVIDIA Isaac Sim 中使用一块 NVIDIA GeForce RTX 4090 GPU 训练约 10 小时。机器人最大速度设为 2.0 m/s。仿真实验在 RTX 4090 台式机上进行，实物飞行的计算则在四旋翼机载计算机（NVIDIA Jetson Orin NX）上完成。使用 Intel RealSense D435i 相机进行静态与动态障碍物感知，并采用激光雷达惯性里程计（LiDAR Inertial Odometry, LIO）算法 [47] 实现精确的机器人状态估计。静态与动态特征提取器均为 3 层卷积神经网络，输出嵌入维度分别为 128 和 64。策略网络为两层多层感知机，PPO 裁剪比例为 0.1。优化器采用 ADAM，学习率为 5×10^{-4}，奖励折扣因子设为 0.99。

A. RL 训练结果

本框架在 RL 策略训练中采用课程学习策略，初始与最终训练环境如图 6 所示。静态障碍物以红色显示，颜色渐变表示高度；动态障碍物以绿色显示。据观察，训练中动态障碍物比静态障碍物更具挑战性。因此，每当导航成功率（定义为从起点无碰撞安全到达目标）超过 80% 时，将环境中的动态障碍物数量以 20 为步长从 60 逐步增加到 120。为验证课程学习的有效性，表 I 在总训练时间相同的条件下，比较了使用与不使用课程学习时达到的最高导航成功率。结果显示，随着动态障碍物数量增加，不使用课程学习时的最高成功率下降更快，体现了课程学习的有效性。策略训练在动态障碍物数为 120 时停止，最终保存的



(a) Initial Training Environment (b) Final Training Environment

图 6. 机器人训练环境可视化。环境大小为 50m×50m，训练过程中动态障碍物数量逐步增加。(a) 初始环境包含 60 个动态障碍物。(b) 最终环境包含 120 个动态障碍物。

表 I
训练中使用与不使用课程学习时的最高导航成功率对比。

障碍物数量	无课程学习	课程学习
static=350, dynamic=60	94.33%	94.33%
static=350, dynamic=80	74.51%	82.71%
static=350, dynamic=100	62.30%	80.96%
static=350, dynamic=120	54.98%	68.65%

是在 100 个动态障碍物下训练的最佳模型，它在高难度条件下取得了较高的成功率（80.96%）。

为说明使用更多机器人训练的重要性，图 8 比较了训练中部署不同数量机器人时的平均 RL 训练回报。结果表明，使用更多机器人训练不仅收敛更快，回报也更高。本文实验使用 1024 个机器人，占满了可用 GPU 显存。

B. 仿真实验

本文在 Gazebo 中开展仿真实验以评估框架性能，并验证其仿真到仿真（Sim-to-Sim）的迁移能力。定量测试在类室内动态环境中进行，这些环境模拟机器人需要安全绕开行人的真实场景。图 7 展示了一条安全轨迹示例。结果证实机器人成功完成了无碰撞导航。

为定量评估性能，本文生成与训练环境条件相当的高障碍物密度环境，其中静态与动态障碍物随机放置。由于开源的 RL 导航基准算法较少，本文与流行的基于优化的静态规划器 [48] 以及面向动态环境的视觉辅助规划器 [20] 进行比较。此外，还评估了本框架在有无安全护盾情况下的性能，以验证安全护盾的有效性。具体而言，在静态、纯动态和混合三类环境中测试，每种算法在每类环境中运行 20 次，计算每次运行的平均碰撞次数，结果见表 II。表 II 中的百分比表示相对于基线方法 [20] 的平均碰撞次数。需要强调，这些环境旨在测试算法极限，复杂度远高于真实场景，因此出现碰撞并不一定意味着算法不安全。总体而言，NavRL 在动态与混合环境中碰撞次数最少，在静态环境中与 EGO 规划器的碰撞率相当。EGO 规划器在动态与混合环境中记为 N/A，原因是其地图更新效率低，处理动态障碍物带来的大量噪声时会卡住。实验中，当障碍物距离机器人很近时，ViGO 无法生成足够灵敏的反应式轨迹进行避碰；相比之下，本方法能生成更灵敏的控制，使机器人更高效地避碰。对比有无安全护盾的结果发现，安全护盾始终能减少碰撞，在动态环境中尤为明显——动态环境下神经网络失效风险更高，安全护盾有效缓解了由此产生的错误。

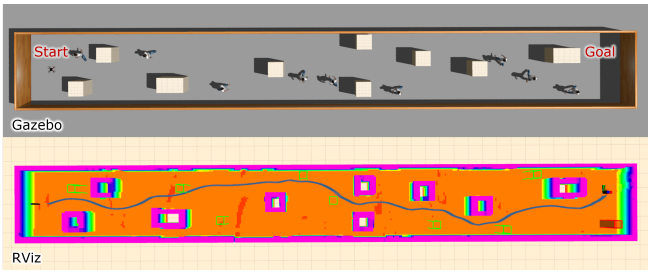


图 7. 仿真走廊环境中安全导航轨迹的可视化。上图为 Gazebo 中的环境，包含静态障碍物与动态障碍物（行人）；下图为环境地图及机器人导航轨迹。

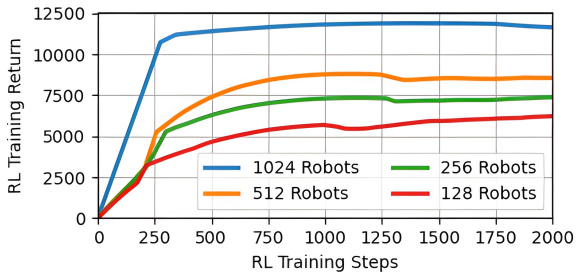


图 8. 平滑后的平均训练回报曲线。使用更多机器人训练可获得更快的收敛与更高的回报。

C. 实物飞行测试

为验证仿真到现实迁移与安全导航能力，本文在多种场景中开展了实物飞行实验，如图 9 所示。环境中放置了静态障碍物，并安排若干行人朝机器人走来，要求机器人在到达目标的同时避开碰撞。结果表明，机器人成功避开碰撞并安全到达目的地。

在仿真与真实飞行实验中，本文测量了框架各模块的计算时间，详见表 III。表中给出了在 NVIDIA GeForce RTX 4090 与机载 NVIDIA Jetson Orin NX 上的运行时间。静态感知模块在 RTX 4090 上耗时 8 ms，在 Orin NX 上耗时 15 ms；动态感知模块分别耗时 11 ms 和 27 ms。RL 策略网络在 RTX 4090 上运行 1 ms，在 Orin NX 上运行 7 ms；安全护盾模块分别为 2 ms 和 16 ms。这些结果表明，即使在机载计算机上，所有模块也都能实时运行。

V. 结论与未来工作

本文提出基于近端策略优化（PPO）算法的深度强化学习框架 NavRL，用于实现动态环境中的安全飞行。该框架采用定制的状态与动作表示，在静态与动态环境中均能安全导航，并支持有效的零样本仿真到仿真及仿真到现实迁移。基于速度障碍概念的安全护盾缓解了神经网络黑盒特性导致的失效；基于 NVIDIA Isaac Sim 的并行训练流程加速了学习过程。仿真与实物实验结果验证了本方法在动态环境中实现安全导航的有效性。未来工作将致力于改进该框架，并将其适配部署到多种机器人平台上。

参考文献

[1] B. Zhou, Y. Zhang, X. Chen, and S. Shen, “Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 779–786, 2021.

表 II
不同类型环境中 20 次运行的平均碰撞次数基准对比。

20m×40m 地图中的平均碰撞次数			
方法	静态环境	动态环境	混合环境
EGO [48]	0.45 (56.3%)	N/A	N/A
ViGO [20]	0.80 (100%)	3.15 (100%)	4.40 (100%)
本文（无安全护盾）	0.95 (118.8%)	2.70 (85.7%)	4.60 (104.5%)
本文（NavRL）	**0.65** (81.3%)	**0.85** (27.0%)	**2.10** (47.8%)



图 9. 实物飞行测试示例。本框架使机器人能够在静态与动态障碍物并存的情况下安全飞行与导航。

表 III
系统各组件的运行时间。

系统模块	GeForce RTX 4090	Jetson Orin NX
静态感知	8 ms	15 ms
动态感知	11 ms	27 ms
RL 策略网络	1 ms	7 ms
安全护盾	2 ms	16 ms

[2] S. H. Alsamhi, A. V. Shvetsov, S. Kumar, S. V. Shvetsova, M. A. Alhartomi, A. Hawbani, N. S. Rajput, S. Srivastava, A. Saif, and V. O. Nyangaresi, "Uav computing-assisted search and rescue mission framework for disaster and harsh environment mitigation," *Drones*, vol. 6, no. 7, p. 154, 2022.

[3] Z. Xu, B. Chen, X. Zhan, Y. Xiu, C. Suzuki, and K. Shimada, "A vision-based autonomous uav inspection framework for unknown tunnel construction sites with dynamic obstacles," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4983–4990, 2023.

[4] M. Kulkarni and K. Alexis, "Task-driven compression for collision encoding based on depth images," in *International Symposium on Visual Computing*. Springer, 2023, pp. 259–273.

[5] D. Hoeller, L. Wellhausen, F. Farshidian, and M. Hutter, "Learning a state representation and navigation in cluttered and dynamic environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5081–5088, 2021.

[6] Y. Song, K. Shi, R. Penicka, and D. Scaramuzza, "Learning perception-aware agile flight in cluttered environments," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1989–1995.

[7] D. Gandhi, L. Pinto, and A. Gupta, "Learning to fly by crashing," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3948–3955.

[8] N. Smolyanskiy, A. Kamenev, J. Smith, and S. Birchfield, "Toward low-flying autonomous mav trail navigation using deep neural networks for environmental awareness," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4241–4247.

[9] V. Tolani, S. Bansal, A. Faust, and C. Tomlin, "Visual navigation among humans with optimal control as a supervisor," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2288–2295, 2021.

[10] A. Ray, J. Achiam, and D. Amodei, "Benchmarking safe exploration in deep reinforcement learning," *arXiv preprint arXiv:1910.01708*, vol. 7, no. 1, p. 2, 2019.

[11] B. Thananjeyan, A. Balakrishna, S. Nair, M. Luo, K. Srinivasan, M. Hwang, J. E. Gonzalez, J. Ibarz, C. Finn, and K. Goldberg, "Recovery rl: Safe reinforcement learning with learned recovery zones," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4915–4922, 2021.

[12] K.-C. Hsu, A. Z. Ren, D. P. Nguyen, A. Majumdar, and J. F. Fisac, "Sim-to-lab-to-real: Safe reinforcement learning with shielding and generalization guarantees," *Artificial Intelligence*, vol. 314, p. 103811, 2023.

[13] N. Kochdumper, H. Krasowski, X. Wang, S. Bak, and M. Althoff, "Provably safe reinforcement learning via action projection using reachability analysis and polynomial zonotopes," *IEEE Open Journal of Control Systems*, vol. 2, pp. 79–92, 2023.

[14] F. Sadeghi and S. Levine, "Cad2rl: Real single-image flight without a single real image," *arXiv preprint arXiv:1611.04201*, 2016.

[15] L. Xie, S. Wang, A. Markham, and N. Trigoni, "Towards monocular vision based obstacle avoidance through deep reinforcement learning," *arXiv preprint arXiv:1706.09829*, 2017.

[16] R. Brilli, M. Legittimo, F. Crocetti, M. Leomanni, M. L. Fravolini, and G. Costante, "Monocular reactive collision avoidance for mav teleoperation with deep reinforcement learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 12 535–12 541.

[17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[18] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The international journal of robotics research*, vol. 17, no. 7, pp. 760–772, 1998.

[19] Y. Wang, J. Ji, Q. Wang, C. Xu, and F. Gao, "Autonomous flights in dynamic environments with onboard vision," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 1966–1973.

[20] Z. Xu, Y. Xiu, X. Zhan, B. Chen, and K. Shimada, "Vision-aided uav navigation and dynamic obstacle avoidance using gradient-based b-spline trajectory optimization," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1214–1220.

[21] G. Chen, P. Peng, P. Zhang, and W. Dong, "Risk-aware trajectory sampling for quadrotor obstacle avoidance in dynamic environments," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 12, pp. 12 606–12 615, 2023.

[22] L. Tai, S. Li, and M. Liu, "A deep-network solution towards model-less obstacle avoidance," in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2016, pp. 2759–2764.

[23] R. P. Padhy, S. Verma, S. Ahmad, S. K. Choudhury, and P. K. Sa, "Deep neural network for autonomous uav navigation in indoor corridor environments," *Procedia computer science*, vol. 133, pp. 643–650, 2018.

[24] A. Loquercio, A. I. Maqueda, C. R. Del-Blanco, and D. Scaramuzza, "Dronet: Learning to fly by driving," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1088–1095, 2018.

[25] S. Jung, S. Hwang, H. Shin, and D. H. Shim, "Perception, guidance, and navigation for indoor autonomous drone racing using deep learning," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2539–2544, 2018.

[26] S. Lv, Y. Gao, J. Che, and Q. Quan, "Autonomous drone racing: time-optimal spatial iterative learning control within a virtual tube," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3197–3203.

[27] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "Vint: A foundation model for visual navigation," *arXiv preprint arXiv:2306.14846*, 2023.

[28] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine, "Gnm: A general navigation model to drive any robot," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7226–7233.

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

[30] N. Simon and A. Majumdar, "Mononav: Mav navigation via monocular depth estimation and reconstruction," in *International Symposium on Experimental Robotics*. Springer, 2023, pp. 415–426.

[31] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller, "Zoedepth: Zero-shot transfer by combining relative and metric depth," *arXiv preprint arXiv:2302.12288*, 2023.

[32] P. Roth, J. Nubert, F. Yang, M. Mittal, and M. Hutter, "Viplanner: Visual semantic imperative learning for local navigation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5243–5249.
[33] A. Singla, S. Padakandla, and S. Bhatnagar, "Memory-based deep reinforcement learning for obstacle avoidance in uav with limited environment knowledge," *IEEE transactions on intelligent transportation systems*, vol. 22, no. 1, pp. 107–118, 2019.
[34] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 285–292.
[35] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1343–1350.
[36] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3052–3059.
[37] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, "Champion-level drone racing using deep reinforcement learning," *Nature*, vol. 620, no. 7976, pp. 982–987, 2023.
[38] L. Bauersfeld, E. Kaufmann, and D. Scaramuzza, "User-conditioned neural control policies for mobile robotics," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1342–1348.
[39] A. Romero, Y. Song, and D. Scaramuzza, "Actor-critic model predictive control," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 14 777–14 784.
[40] J. Xing, L. Bauersfeld, Y. Song, C. Xing, and D. Scaramuzza, "Contrastive learning for enhancing robust scene transfer in vision-based agile flight," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 5330–5337.
[41] B. Xu, F. Gao, C. Yu, R. Zhang, Y. Wu, and Y. Wang, "Omnidrones: An efficient and flexible platform for reinforcement learning in drone control," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2838–2844, 2024.
[42] T. He, C. Zhang, W. Xiao, G. He, C. Liu, and G. Shi, "Agile but safe: Learning collision-free high-speed legged locomotion," in *Robotics: Science and Systems (RSS)*, 2024.
[43] Z. Xu, X. Zhan, Y. Xiu, C. Suzuki, and K. Shimada, "Onboard dynamic-object detection and tracking for autonomous robot navigation with rgb-d camera," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 651–658, 2024.
[44] H. Oleynikova, D. Honegger, and M. Pollefeys, "Reactive avoidance using embedded stereo vision for mav flight," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 50–56.
[45] P.-W. Chou, D. Maturana, and S. Scherer, "Improving stochastic policy gradients in continuous control with deep reinforcement learning using the beta distribution," in *International conference on machine learning*. PMLR, 2017, pp. 834–843.
[46] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research: The 14th International Symposium ISRR*. Springer, 2011, pp. 3–19.
[47] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.
[48] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, "Ego-planner: An esdf-free gradient-based local planner for quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 478–485, 2020.