基于 DQN 的无人机自主导航技术报告

ZY2557445 汪谨遵

1 引言

本项目实现了基于深度 Q 网络 (Deep Q-Network, DQN) 的无人机自主导航系统。无人机需要在 AirSim 仿真环境中学习穿越一系列圆形洞口，避免与障碍物碰撞。该任务具有挑战性，要求智能体能够从 RGB 图像中提取空间特征，并做出准确的飞行决策。

2 算法描述

2.1 DQN 算法原理

DQN 是一种结合深度学习与强化学习的算法，通过神经网络近似 Q 函数来学习最优策略。核心思想是最小化时序差分误差：

$$L(\theta) = \mathbb{E}_{(s,a,r,s')\sim D}\left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta)\right)^2\right] \tag{1}$$

其中 θ 为当前网络参数，θ^- 为目标网络参数，D 为经验回放缓冲区，γ 为折扣因子。

2.2 网络架构

本项目采用 VGG16 作为特征提取器，并引入注意力机制增强空间感知能力。网络结构如下：

- **特征提取层**：使用 VGG16 前 23 层卷积层，保留更多空间信息
- **注意力模块**：通过 1×1 卷积生成注意力权重，突出关键区域
- **全连接层**：三层全连接网络 [512, 512, 256]，输出 9 个动作的 Q 值

特征提取器采用预训练权重，前 10 层参数冻结，后续层进行微调，平衡迁移学习与任务适应。

2.3 动作空间

智能体可执行 9 个离散动作，对应无人机在 YZ 平面的移动方向：

动作编号	Y 方向速度	Z 方向速度
0	-0.4	-0.4
1	0	-0.4
2	0.4	-0.4
3	-0.4	0
4	0	0
5	0.4	0
6	-0.4	0.4
7	0	0.4
8	0.4	0.4

表 1: 动作空间定义

无人机 X 方向保持恒定速度 0.4 m/s 前进。

2.4 奖励函数

奖励函数设计综合考虑目标接近度、对齐精度和任务完成情况：

$$R = \begin{cases} 20 \times (\Delta d_{prev} - \Delta d_{curr}) + R_{align} & \text{正常飞行} \\ -100 & \text{碰撞} \\ 10 & \text{成功穿越} \\ -100 & \text{目标丢失} \end{cases} \tag{2}$$

其中 Δd 为无人机与目标中心的距离，R_{align} 为对齐奖励：

$$R_{align} = \begin{cases} 19 & d < 0.30 \text{ 且} x > 2.9 \\ 12 & d < 0.30 \\ 7 & d < 0.45 \\ 0 & \text{其他} \end{cases} \tag{3}$$

3 实验设置

3.1 环境配置

- **仿真平台**：Microsoft AirSim + Unreal Engine 4
- **观测空间**：$50 \times 50 \times 3$ RGB 图像
- **仿真加速**：20 倍速（ClockSpeed=20）
- **洞口间距**：4 米
- **洞口半径**：0.3 米

3.1.1 训练环境与测试环境

本项目采用不同的环境配置用于训练和测试：

- **训练环境（TrainEnv）**：使用特定的场景纹理和洞口位置分布进行训练
- **测试环境（TestEnv）**：采用与训练环境不同的纹理和洞口位置分布，用于评估模型的泛化能力

这种设计能够真实评估模型在未见过的环境中的泛化性能和鲁棒性。

3.2 训练参数

参数	数值
总训练步数	500,000
学习率	3×10^{-4}
批次大小	64
经验回放缓冲区	100,000
折扣因子 γ	0.99
探索率衰减	线性衰减 (1.0→0.02)
目标网络更新频率	10,000 步
评估频率	1,000 步 (0-390K) / 10,000 步 (390K-500K)
优化器	Adam

表 2: DQN 训练超参数

评估频率调整说明：训练初期采用较高的评估频率（每 1,000 步），以密切监控模型性能。但频繁评估显著降低了训练速度，前 390K 步耗时超过 1 天。因此在 390K 步后将评估频率降低至 10,000 步，在保证性能监控的同时大幅提升训练效率。

3.3 硬件环境

- **GPU**：NVIDIA RTX 5060
- **框架**：PyTorch 1.7.1, Stable-Baselines3 1.2.0
- **总训练时间**：约 3 天（含评估频率优化前后和第二阶段训练）

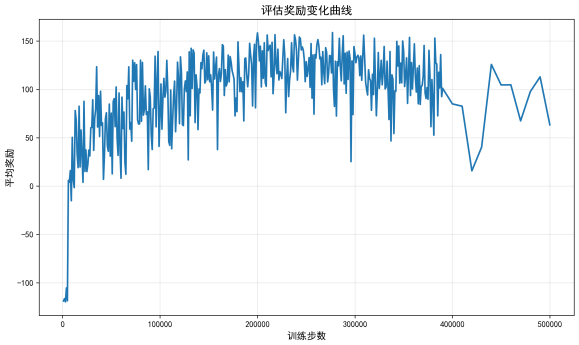
4 实验结果

4.1 单洞训练（第一阶段）

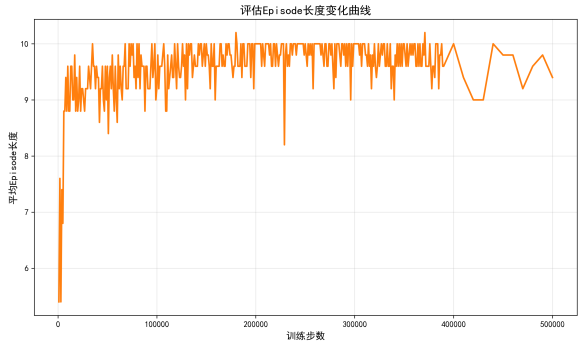
第一阶段采用标准 DQN 训练，每次穿越一个洞后重置环境。训练过程中，模型性能稳步提升。图1展示了训练过程中的关键指标变化。

从图1可以看出：

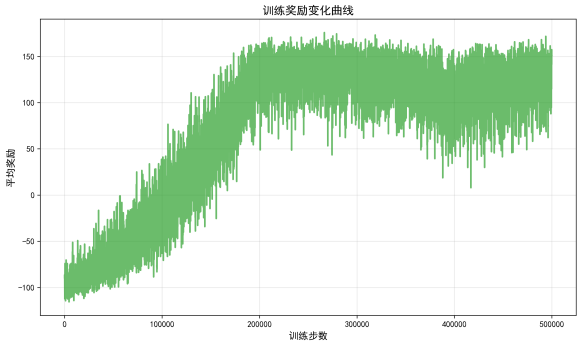
- **评估奖励**：从初始的负值逐步提升至 150 之间，并在后期保持稳定



(a) 评估奖励变化



(b) Episode 长度变化



(c) 训练奖励变化

图 1: 单洞训练过程关键指标变化曲线

- **Episode 长度**：从初始的 0 步左右提升至 10-11 步，表明智能体学会了更精确的导航
- **训练奖励**：整体呈上升趋势，验证了学习过程的有效性
- **收敛性**：约在 300K 步后模型性能趋于稳定，表明训练充分

使用最佳模型在测试环境（TestEnv）进行 177 个 episode 的测试，结果如下：

指标	数值
总测试 episodes	177
平均飞行距离	25.67 m
最大飞行距离	102.74 m
平均穿洞数	6
最大穿洞数	25
单洞成功率	60-70%

表 3: 单洞训练模型测试结果（TestEnv）

4.2 多洞训练（第二阶段）

为进一步提升连续穿洞能力，进行了第二阶段训练。该阶段采用迁移学习策略，加载第一阶段最佳模型权重，在训练环境（TrainEnv）中继续训练。与单洞训练不同，多

洞训练允许无人机在成功穿越一个洞后继续前进挑战下一个洞，直到发生碰撞才结束episode。

4.2.1 训练配置

参数	数值
总训练步数	600,000
学习率	1×10^{-4}
初始探索率	0.2
预训练模型	第一阶段 best_model
环境类型	TrainEnv
奖励函数	连续穿洞累积奖励

表 4: 多洞训练超参数

多洞环境的奖励函数设计为：

$$R_{multi} = R_{base} + \begin{cases} 50 + n \times 10 & \text{穿越第}n\text{个洞} \\ -100 & \text{碰撞} \end{cases} \tag{4}$$

其中 R_{base} 为基础导航奖励，n 为已穿越洞口数量。

4.2.2 训练过程

图2展示了多洞训练的评估奖励变化曲线。

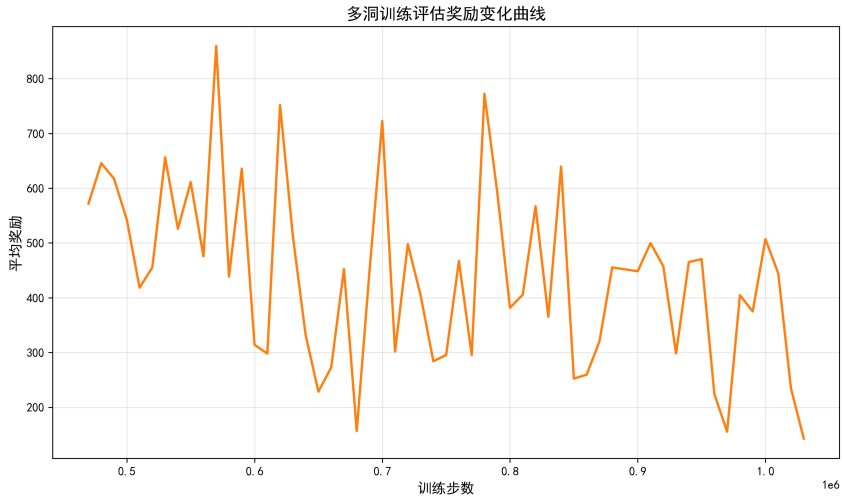


图 2: 多洞训练评估奖励变化曲线

4.2.3 测试结果

使用多洞训练的最佳模型在测试环境（TestEnv）进行 709 个 episode 的测试：

指标	数值
总测试 episodes	709
平均飞行距离	18.40 m
最大飞行距离	43.16 m
平均穿洞数	4
最大穿洞数	10

表 5: 多洞训练模型测试结果（TestEnv）

4.3 两阶段对比分析

4.3.1 性能对比

图3展示了单洞训练与多洞训练的评估奖励对比。

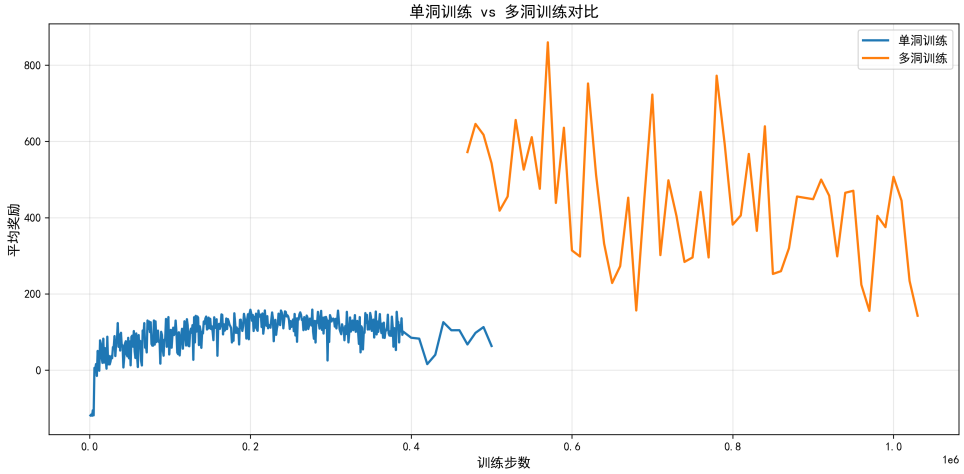


图 3: 单洞训练 vs 多洞训练评估奖励对比

指标	单洞训练	多洞训练
平均飞行距离	25.67 m	18.40 m
最大飞行距离	102.74 m	43.16 m
平均穿洞数	6	4
最大穿洞数	25	10

表 6: 两阶段训练结果对比

对比结果显示，多洞训练模型的性能反而下降。平均穿洞数从 6 降至 4，最大穿洞数从 25 降至 10。这一反直觉的结果值得深入分析。

4.4 性能下降原因分析

通过对比图3中的训练曲线和测试结果，可以从以下几个方面分析多洞训练性能下降的原因：

1. 过拟合问题

从图3可以看出，多洞训练的评估奖励在训练环境中持续上升，但在测试环境中性能却显著下降。这是典型的过拟合现象。由于训练环境和测试环境具有不同的纹理和洞口位置分布，多洞训练使模型过度拟合训练环境的特定视觉特征和空间布局，导致模型失去了对新环境的泛化能力。

单洞训练虽然也在同一训练环境中进行，但通过频繁的环境重置和随机起始位置，引入了更多的样本多样性，使模型学习到更加通用的视觉特征和导航策略。这种训练方式降低了对特定环境特征的依赖，提高了模型的泛化能力。

2. 奖励函数设计缺陷

多洞环境的累积奖励机制存在以下问题：

- **奖励稀疏**：模型需要连续穿越多个洞才能获得显著奖励，增加了学习难度
- **奖励延迟**：穿洞奖励的累积特性导致信用分配问题，模型难以判断哪些动作真正有效
- **保守策略**：高额的碰撞惩罚 (-100) 可能引导模型采取过于保守的策略，优先避免碰撞而非追求更多穿洞

相比之下，单洞训练的奖励函数更加平衡，既鼓励接近目标，又给予穿洞即时奖励，使学习过程更加高效。

3. 探索-利用失衡

第二阶段将初始探索率降至 0.2，虽然有利于利用已学知识，但在更复杂的多洞环境中，这一设置可能限制了对新策略的探索。从图3可以看出，多洞训练曲线的波动较大，说明模型在探索和利用之间未能找到良好平衡。

单洞训练从 1.0 线性衰减至 0.02 的探索策略，为模型提供了充分的探索空间，使其能够发现更优的导航策略。

4. 灾难性遗忘

迁移学习过程中，模型在适应新任务时可能遗忘了第一阶段学到的基础导航技能。虽然采用了较低的学习率 (1×10^{-4}) 来保护已学知识，但在 600K 步的长时间训练中，仍无法完全避免遗忘现象。

这一问题在图3中表现为多洞训练曲线在后期出现波动，说明模型在新旧知识之间产生了冲突。

5. 训练-测试环境不匹配

如前文所述，训练环境（TrainEnv）和测试环境（TestEnv）具有不同的纹理和洞口位置分布。这种环境差异对模型的泛化能力提出了挑战。

多洞训练通过长时间的连续穿越，使模型过度拟合训练环境的特定纹理特征和洞口位置模式。当面对测试环境中不同的视觉特征和空间布局时，模型难以适应，导致性能显著下降。

相比之下，单洞训练通过频繁的环境重置和随机起始位置，虽然在同一训练环境中进行，但增加了样本多样性，使模型学习到更加鲁棒的视觉特征提取能力。这种训练方式使模型在面对测试环境的新纹理和新布局时，仍能保持较好的性能。因此单洞训练模型的泛化能力更强。

4.5 模型选择

综合考虑性能和泛化能力，选择第一阶段的 `best_model.zip` 作为最终部署模型。该模型在单洞训练中学到的基础导航技能更加稳定，泛化能力更强，在实际测试中表现出更优的性能。

5 技术亮点

5.1 特征提取优化

1. **迁移学习**：使用 ImageNet 预训练的 VGG16 权重，加速收敛
2. **注意力机制**：引入空间注意力模块，提升对洞口位置的感知能力
3. **分层微调**：冻结浅层特征，仅微调深层特征，防止过拟合

5.2 训练策略

1. **经验回放**：打破样本相关性，提高样本利用效率
2. **目标网络**：稳定训练过程，避免 Q 值估计震荡
3. **评估驱动**：定期评估并保存最佳模型，确保泛化性能
4. **断点续训**：支持训练中断后自动恢复，提高训练灵活性

6 结论

本项目成功实现了基于 DQN 的无人机自主导航系统。通过结合 VGG16 特征提取器和注意力机制，模型能够从 RGB 图像中学习有效的导航策略。

实验进行了两个阶段的训练：

1. **单洞训练**：模型学习基础穿洞技能，平均穿洞数 6 个，最大 25 个
2. **多洞训练**：尝试通过迁移学习提升连续穿洞能力，但出现性能下降

多洞训练的性能下降揭示了强化学习中的几个关键问题：过拟合、奖励函数设计、探索-利用平衡、灾难性遗忘等。这些问题在迁移学习和任务泛化中普遍存在，需要更精细的算法设计和训练策略。

最终选择单洞训练的 `best_model.zip` 作为部署模型，该模型展现出更强的泛化能力和稳定性。