论文（设计）题目：

基于深度强化学习的目标导航方法研究

2022 年 05 月 15 日

摘　要

自主导航是机器人完成许多其他任务的基本能力要求，近年来兴起的深度强化学习为机器人的自主目标导航提供了一种解决思路，但是现有的基于深度强化学习的视觉目标导航方法存在跨场景泛化能力差等问题。本文从提高跨场景泛化能力入手，参考前人的思路并提出了一种具有较强跨场景泛化能力的端到端的视觉目标导航模型。

针对跨场景泛化能力弱的问题，本文提出了一种结合目标检测结果和深度图的状态表示方式以及一种结合目标检测结果的奖励函数表示方法。由于状态表示中包含的场景特有信息较少，因此将该种状态表示方法和奖励函数表示方法相结合保证模型既能拥有较强的跨目标泛化能力，也能拥有较强的跨场景泛化能力。

此外，本文将 AI2THOR 仿真场景制作成了离线 AI2THOR 数据集，相较于直接使用 AI2THOR 仿真平台实时渲染，使用离线 AI2THOR 数据集能大大提高交互速度，并且能实现仿真场景的跨平台使用。在此数据集的基础上，本文对所提出的导航模型进行了训练和实验。实验结果表明，所提出的模型有更好的导航表现，并且在跨目标泛化能力和跨场景泛化能力上都有所提高。

关键词：深度强化学习；视觉目标导航；跨目标泛化能力；跨场景泛化能力

ABSTRACT

Autonomous navigation is the basic ability requirement for robots to complete many other tasks. In recent years, the emerging deep reinforcement learning provides a solution for autonomous target navigation of robots, but the existing visual target navigation methods based on deep reinforcement learning have some problems such as poor cross-scene generalization ability.

In this paper, an end-to-end visual object navigation model with strong ability of cross-scene generalization is proposed based on previous ideas. To solve the problem of weak cross-scene generalization, this paper proposes a state representation method combining target detection results with depth image and a reward function representation method combining target detection results. Since the state representation contains less scene-specific information, the combination of the state representation and the reward function representation ensures that the model can have strong cross-objective generalization ability as well as cross-scene generalization ability.

In addition, the AI2THOR simulation scene is made into the Offline AI2THOR Dataset in this paper. Compared with the direct real-time rendering of AI2THOR simulation platform, the Offline AI2THOR Dataset can greatly improve the interaction speed and realize the cross-platform use of simulation scenes. On the basis of this dataset, the proposed navigation model is trained and experimented. Experimental results show that the proposed model has better navigation performance and has improved cross-target generalization ability and cross-scene generalization ability.

Keywords: deep reinforcement learning; visual target navigation; cross-target generalization ability; cross-scene generalization ability

目　录

第 1 章　绪论

1.1 课题的研究背景与意义

现如今，机器人在人类社会中起的作用越来越大，凭借良好的机动性与强大的续航能力，机器人能帮助人类完成很多任务。例如，在工业领域，机器人能代替人类完成一些危险、繁杂的工作[1]；在服务行业，机器人能代替人类去完成一些琐碎的服务工作[2]；在医疗行业，机器人能和人类一起站到医疗第一线，去协助完成各项艰巨的任务[3]。随着科技发展，机器人在人类社会的细分领域中能完成的任务越来越多，但在这些细分领域中对机器人的要求也越来越高。例如家庭服务机器人不仅要能做家务，还要能跟人类沟通或帮助拾取一些物品等[4]。

为完成这些复杂的任务，自主导航成为机器人需要具备的基础功能[5]。而要能实现自主导航，机器人需要拥有能感知周围环境的能力。由于视觉信息中包含的内容十分丰富，是人类接收环境信息的主要渠道，人们希望机器人也能具备强大的视觉感知能力，而计算机硬件算力的提升和人工智能技术的发展将这一愿望成为可能。现如今，越来越多的研究都聚集到机器人的视觉自主导航上。通过视觉自主导航，机器人能获得更强大的自主导航能力，这将极大地促进机器人在不同应用环境间的部署，使机器人具备更强的实用性[6]。

传统的机器人导航方法，如激光 SLAM[7]或视觉 SLAM[8]，过多地依赖对环境的先验信息，因此进行导航任务之前通常需要先构建环境的地图，或了解环境中各物体的空间布局关系[9]。这不仅导致机器人跨场景导航能力差，机器人在动态环境下的导航也受到很大的影响。强化学习[10]或许是解决这些问题的思路。作为机器学习领域中区别于监督学习和无监督学习的一个重要方向，强化学习是一种基于试错的学习方法。在强化学习中，智能体通过与环境间的交互获取经验并进行学习，逐渐完善自身的策略，进而能在任务当中获得更好的表现[10]。将强化学习应用到导航任务中，机器人能通过与环境间的交互，逐步学习到应该如何根据自身当前的状态采取相应的动作实现避障，进而高效完成导航任务[11]。可见强化学习能减轻机器人导航对环境先验信息的依赖，使机器人在导航任务中具备更强的自主性，也更符合人们对机器人自主导航的期望。

传统的强化学习方法存在着只能应用到环境状态数量有限且动作空间较小的情况中等问题。近些年来深度学习的蓬勃发展为解决这些问题提供了有力的帮助。人们通过将深度学习和强化学习相结合，形成了一个新的研究热点——深度强化学习(Deep Reinforcement Learning, DRL)[12]。相较于传统的强化学习，深度强化学习有着无可比拟的优势，例如通过使用深度学习的函数逼近器对策略、值

函数等进行近似[13]，能实现从高维的状态空间或动作空间到低维的映射，使强化学习应用的范围得以扩展[14]；例如深度学习中的卷积神经网络能直接输入视觉图像，将卷积神经网络与强化学习结合，使端到端训练与控制成为可能[15]。

然而由于深度学习和强化学习本身的一些缺陷，使深度强化学习依然存在很多问题，例如数据利用率低、泛化能力差等[14]，而这些问题也成为如今深度强化学习的研究重点，也是实现端到端的机器人导航功能的关键。

1.2 国内外研究现状

1.2.1 深度强化学习研究现状

深度强化学习集成了深度学习在视觉等感知问题上强大的理解能力以及强化学习的决策能力，实现了端到端学习。深度强化学习的出现使强化学习技术真正走向实用，能解决现实场景下的复杂问题。

2015 年，Mnih 等在《自然》杂志上发表论文[1]，提出了一个结合深度学习技术和强化学习思想的模型深度 Q 网络(Deep Q-Network, 简称 DQN)，在 Atari 游戏平台上展现出超越人类水平的表现[17]。此后，DeepMind 公司将强化学习应用到围棋领域，相继推出围棋智能体 AlphaGo[18]和 AlphaZero[19]，并使用它们多次战胜人类围棋冠军。这些成功使深度强化学习成为人工智能界关注的热点。

与此同时，很多优秀的 DRL 算法也相继被提出。DeepMind 公司将 DQN 与确定性策略梯度算法结合，推出深度确定性策略(Deep Deterministic Policy Gradient, 简称 DDPG)算法[20]；由于强化学习得到的策略通常是一个单峰分布，但是在实际应用中解决方法通常不止一个，Haarnoja 在 DDPG 算法的基础上结合柔性 Q 学习[21]思想，推出柔性演员-评论家算法(Soft Actor-Critic, 简称 SAC)[22]，并在机器人仿真任务上取得成功；为解决强化学习算法存在的训练不稳定问题，Schulman 提出信赖域策略优化算法(Trust Region Policy Optimization, 简称 TRPO)[23]，通过限制模型更新时新旧策略相对熵大小，保证模型性能随着更新单调递增；此外，Schulman 为降低 TRPO 算法的计算复杂度，提出近端策略优化算法(Proximal Policy Optimization, 简称 PPO)[24]，引入代理损失函数，在降低算法计算复杂度的同时还保证相同的效果。

为验证深度强化学习算法效果，推进深度强化学习算法应用，许多相关的仿真平台也被开发出来。在游戏方面，除经典的 Atari 等小型游戏外，业界同样开放了包括王者荣耀[25]、星际争霸[26]等大型游戏的接口供强化学习算法训练；在机器人领域，不仅有 AI2THOR[27-29]、Habitat[30-31]等室内环境仿真平台，还有 Mujoco[32]、RLBench[33]等方便仿真各种机器人运动学规律的平台；在工程领域，

Facebook 提出开源强化学习平台 Horizon[34]，用来优化大规模生产系统。

结合仿真平台，深度强化学习算法被用来解决许多实际问题。在工业领域，UC Berkeley 的学者 Sergey 使用深度强化学习算法训练机械臂，使其能完成简单的物体抓取任务[35]；在自动驾驶领域，许多工作应用深度强化学习算法使汽车具备自主导航功能[36]；在游戏领域，天津大学的学者 Haotian Fu 与网易公司合作，利用深度强化学习算法训练出强大的智能体，并将其应用到多款游戏中[37-38]。

1.2.2 基于视觉的导航方法研究现状

研究表明，人类获取的信息中有 83%来源于视觉[39]，在机器人领域，视觉输入也是机器人获取外界信息的主要途径。与超声波传感器等其他传感器获取的信息相比，视觉信息包含更丰富的内容。目前，通过对视觉信息的提取与利用，机器人借助视觉完成的任务越来越多，例如目标检测等。基于视觉的导航方法作为近些年来机器人领域的热点，得到了人们的关注。如何高效地从视觉信号提取出有助于目标导航的信息，是实现机器人高效进行目标导航的关键[40]。

传统的基于强化学习的导航方法在训练时通常设定智能体的目标点不变，但该训练方式往往会带来智能体的跨目标泛化能力不足的问题。2016 年，斯坦福的学者 Y.Zhu 等人[41]经过分析，认为造成这一问题的原因是模型在学习过程中将导航目标隐式地编码到模型参数中，导致更换目标后表现不佳。

为解决该问题，Y.Zhu 等人将导航目标的 RGB 图像输入模型，让模型减弱对训练过程中设定的导航目标的依赖。最终，作者将模型应用到 AI2THOR[27]仿真平台，实验表明，相较于传统的视觉目标导航方法，该导航方法具有在同一场景下较好的跨目标泛化能力，可以说是基于视觉的目标导航方法领域的进步。然而，实验也表明，模型的跨场景泛化能力存在较大的提升空间。

2018 年，Google 的学者 A.Mousavian 等人深入探讨使用不同形式的视觉表达对导航模型性能的影响[43]。他们将深度图、原始 RGB 图像、目标检测结果图以及语义分割结果作为同一个模型的不同视觉表达方式。实验结果表明，利用目标检测结果以及语义分割结果等高级视觉信息作为模型的视觉表达方式能让模型在泛化能力以及从仿真到现实的迁移能力等方面均取得一定的提升。这是因为使用高级语义信息作为视觉表达方式能让模型对不同场景间、虚拟与真实间的差异不敏感，以获得更强的适应能力。例如，目标检测只关注物体的类别与位置，而不关注物体所在场景，因此即使更换到新场景，也能取得不错的导航效果。

除了跨目标、跨场景泛化能力差以外，将深度强化学习算法应用到机器人目标导航方法中同样存在其他问题——容易过拟合、样本效率低、多智能体交互困难等，这些问题限制了 DRL 在机器人目标导航领域的进一步应用。

2017 年，DeepMind 公司的 Lanctot 等人在一篇文章中指出使用强化学习训

练的模型很容易过拟合到训练的环境中[44]，而深度强化学习作为强化学习的一种，很多学者也提出了很多相关的改进，例如学者 Farebrother 提出将监督学习中的正则化等一些避免过拟合的方法应用到深度强化学习中[45]；华盛顿大学的学者 Wortsman 等人则提出了元强化学习的方法[46]，试图让智能体能具备更强的可拓展性，但训练元强化模型面临着严重的样本效率问题[47]。

此外，样本利用效率低的问题同样也限制了深度强化学习在机器人自主导航方法上的应用。针对该问题，Andrychowicz 等提出事后经验回放算法（Hindsight Experience Replay, 简称 HER）[48]，该算法能提升学习多目标策略时的样本效率；Riedmiller 等人在此基础上提出一种反层扩展的 HER 算法，进而进一步加速了训练[49]。然而由于 HER 是为每一个目标生成样本，并将生成的样本储存在缓存区中的方法，因此 HER 只适用于离策略（Off-Policy）强化学习算法。

最后，如何提高智能体在导航过程中的避障能力以及多智能体间的交互也成为一个研究的方向。MIT 的学者 Chen 等人设计出了一种去中心化的多智能体路径规划框架[50]，并通过长短期记忆(Long Short-Term Memery, 简称 LSTM)网络对环境中的其他智能体的行为进行预测，最终的实验也表明应用了这种框架之后智能体能在动态环境中以人类步行的速度前进。

1.3 论文的主要内容与章节安排

1.3.1 主要内容

本文的主要工作是在 Y.Zhu 等人的工作[41]以及 A.Mousavian 等人的工作[43]基础上，针对现有的机器人视觉目标导航方法存在的跨场景泛化能力弱等问题，提出一种结合目标检测方法和深度图的目标导航方法，该方法的主要思想是降低模型输入中场景特定信息含量，使模型专注于导航决策，提升其跨场景泛化能力。

本文所提出的方法在异步演员-评论员算法（Asynchronous Advantage Actor-Critic, 简称 A3C）[53]的基础上训练智能体完成目标驱动的导航任务。当模型以原始的 RGB 观测图像作为输入时，由于 RGB 观测图像中包含很多场景特有的信息，导致模型在训练过程中记录了很多场景特有信息，从而导致模型跨场景泛化能力下降的问题，因此本工作选用了深度图和目标检测结果这些包含场景特定信息较少的内容作为模型输入，并在 AI2THOR 室内环境仿真平台上对所提出的模型对跨目标泛化能力和跨场景泛化能力等各方面进行性能验证。

1.3.2 章节安排

本文一共分为四章，各章节内容安排如下：

第一章对全文内容进行概述。首先介绍机器人视觉导航的研究背景和研究意义，然后介绍近些年来深度强化学习领域的算法发展情况以及深度强化学习在视觉导航领域的应用，最后对全文主要内容进行介绍，并介绍本文各章节内容安排。

第二章对深度学习以及强化学习领域的基础知识进行详细阐述。首先介绍了深度学习中前向神经网络和卷积神经网络各自的特点和基础理论知识，然后介绍深度神经网络的训练和优化方法，接下来是对强化学习的体系进行详细介绍，最后是对本工作中所用到的异步优势演员-评论员算法，即 A3C 算法进行详细介绍。

第三章主要介绍了本文的工作。首先针对现有方法中模型输入存在的问题进行简单的介绍，然后阐述相应的一些解决方法，最后对本工作的模型表示、终止状态判据、奖励函数设置等方面进行介绍。

第四章主要是对前一章中提出的工作在 AI2THOR 仿真平台上进行实验并对实验结果进行分析。首先介绍了算法的超参数设置、训练环境等细节，然后将本工作提出的模型和 Y.Zhu 等人的工作在跨目标泛化能力与跨场景泛化能力等方面进行比较，接下来是分析不同性能的目标检测器对模型导航表现的影响，最后提出一些与模型相关的问题的讨论以及进一步的工作等。

全文的整体框架如图 1-1 所示。

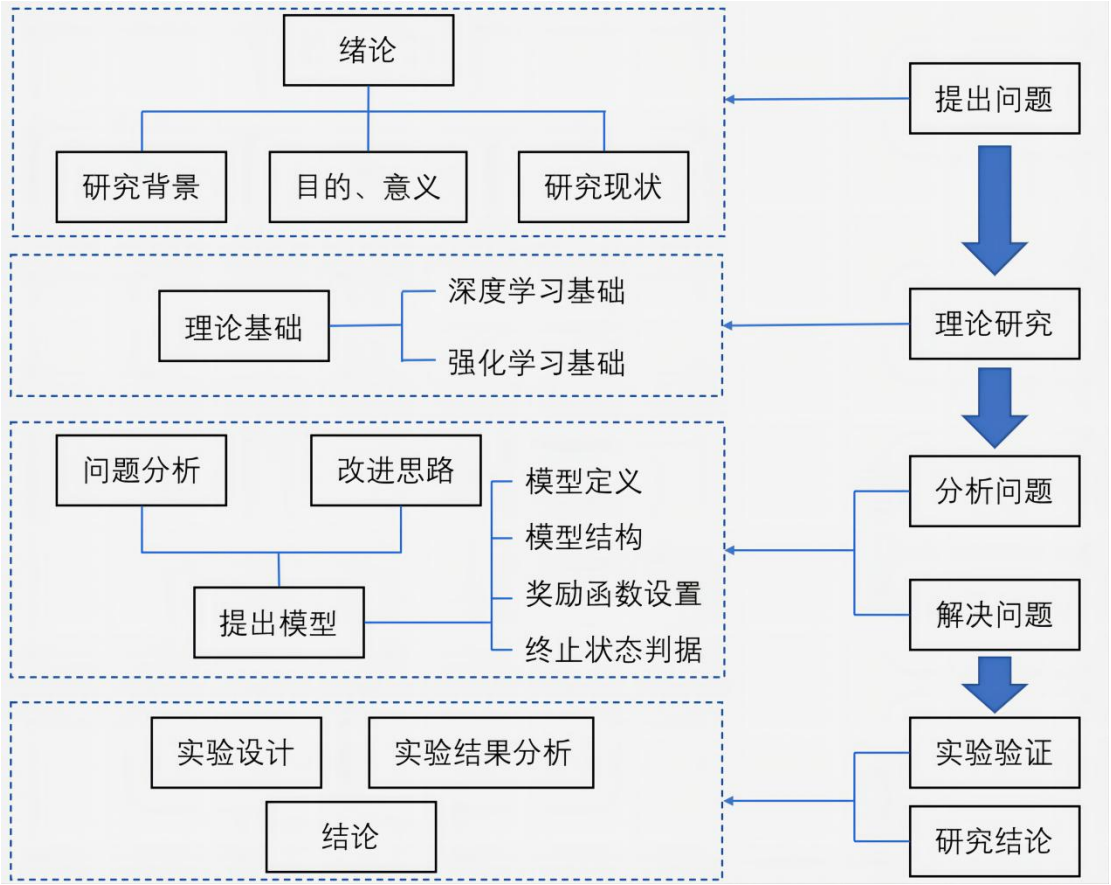

图 1-1 论文框架图

1.4 本章小结

本章主要对课题的研究背景和意义、深度强化学习的研究现状以及基于视觉的目标导航方法研究现状进行了归纳、总结和介绍，最后对本文的章节安排和整体框架进行了梳理。

第 2 章　深度强化学习理论基础

2.1 深度学习基础

深度学习的概念最早由 Hinton 提出[54]，属于机器学习的一个分支。与一般的机器学习算法相比，深度学习能自动进行特征提取，并利用提取到的特征进行学习以解决问题。目前，深度学习已广泛应用于图像处理、语音识别等领域[55]。

2.1.1 全连接神经网络

全连接神经网络是一种常见的前向神经网络，又称多层感知机神经网络（Multi-Layer Perceptron, 简称 MLP）。它通过隐藏单元中的偏置值以及连接不同层隐藏单元间的权重来实现逻辑表达能力。图 2-1 是一个典型的全连接神经网络的结构图，整个神经网络能分为输入层、隐藏层以及输出层三个部分，每一层均由多个神经元组成。全连接神经网络的特点是各层间是全连接的，即相邻两层上的任意两个神经元间均有连接。

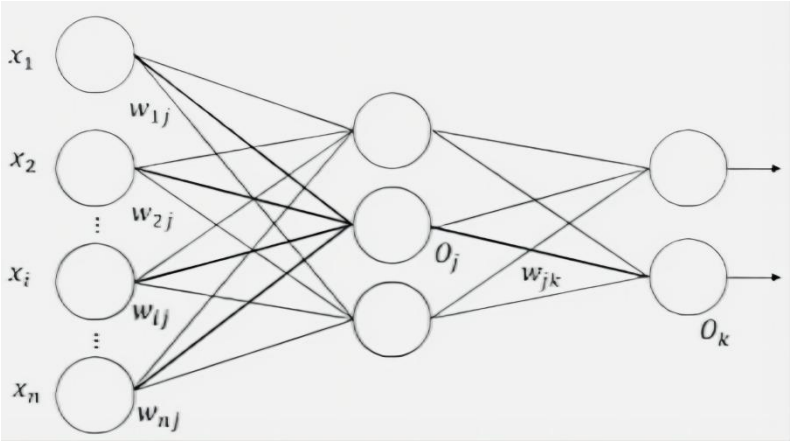

图 2-1 全连接神经网络结构

2.1.2 卷积神经网络

卷积神经网络（Convolutional Neural Networks, 简称 CNN）是另一种典型的前向神经网络。与全连接神经网络不同，卷积神经网络的特点之一是每一层的神经元仅响应上一层的部分神经元，这种特性又被称为稀疏交互。

除了稀疏交互以外，卷积神经网络还有参数共享的特点，也即在进行卷积操作的时候，每一张特征图都使用一个相同的模板对整个图像进行操作。这种参数共享的机制大大减少了卷积神经网络的参数量，使卷积神经网络在训练时需要更新的参数较少，提高了卷积神经网络的训练效率，有利于卷积神经网络规模的扩大，使卷积神经网络能应用于更多的场合当中。

卷积神经网络中有两个常用的基本操作——卷积操作和池化操作。卷积操作的作用是提取图像中的特征，根据采用的卷积核参数的不同，卷积操作能起到提取边缘、滤波等不同作用；池化操作能分为平均池化和最大池化，其作用是缩小参数矩阵的尺寸，减少特征图的参数量，提高训练效率并防止卷积神经网络过拟合。

如图 2-2 所示，在图像分类等任务中，通常在卷积神经网络之后连接上一些全连接层，组成深度卷积神经网络。深度卷积神经网络能处理更为复杂的输入信息，拥有更强的表达能力，但是引入的全连接层大大增加了模型的参数量，使模型消耗的计算资源更多。

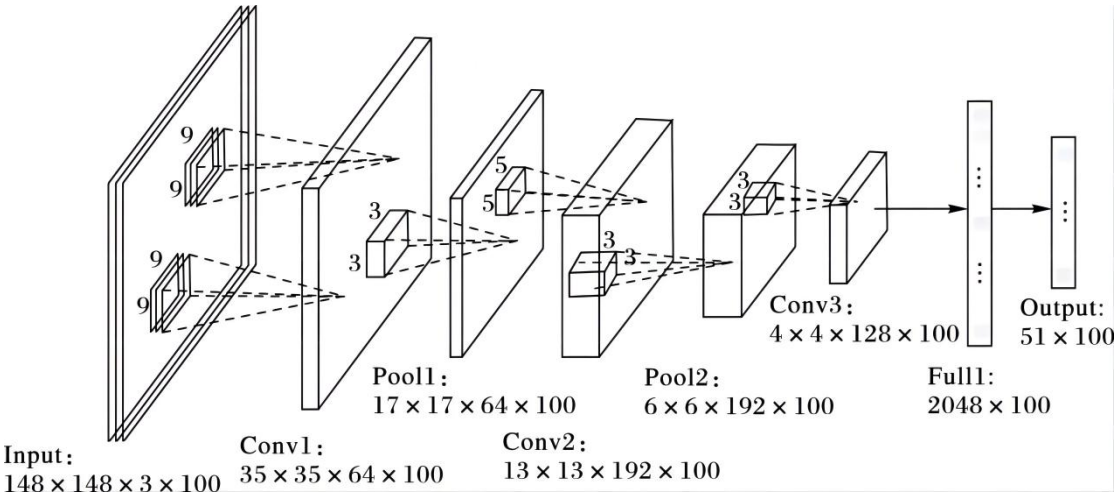

图 2-2 深度卷积神经网络

2.1.3 深度神经网络的训练

在使用深度神经网络时，通常将数据由输入层送入，数据经过中间各个隐藏层的节点并产生输出，该过程被称为前向传播；而在网络训练时，则一般使用反向传播算法（Back Propagation，简称 BP）[56]。在监督学习中，数据集中的各个数据都会有一个对应的标签，人们为数据通过深度神经网络前向传播得到的输出值和数据的标签设计一个损失函数。在训练过程中，将损失函数的信息沿着深度神经网络的各个层向后传播，并计算出相应的梯度，通过梯度下降算法即可对网络中的参数——权重和偏置进行更新，实现对深度神经网络进行优化的目的。

梯度下降算法是一种通过迭代的方式来寻找最优参数的算法。通过多次迭代，梯度下降算法会使损失函数到达局部最优解或最小值，然而人们通常希望损失函数最后能到达最小值而并非局部最优解，因此对传统的梯度下降法进行了优化，提出了动量法等方法使损失函数在迭代时能冲出局部最优解，继续优化。此外，由于在训练时梯度下降算法会在一次迭代中使用训练集中的全部样本，这导致迭代效率较低，每次迭代都需要消耗大量的资源。为解决这个问题，人们提出了随

机梯度下降算法，也即每次迭代时仅使用训练集中的一个样本。但是由于该方法仅使用一个样本，迭代时带来的随机性较大，导致网络训练的时间变长。因此人们综合梯度下降法和随机梯度下降法，提出了小批量随机梯度下降法，即每次迭代时从训练集中选取一部分样本进行训练，既保证了较高的网络更新的效率，又能保证较短的网络训练时间。

随着神经网络的不断加深，梯度爆炸和梯度消失的问题引起了人们的注意。由于神经网络在反向传播过程中参数更新的方法遵循链式法则，在链式求导公式中，若导数项大于 1，随着网络层数的增加，就容易发生梯度爆炸；若导数项小于 1，随着网络层数的增加则容易发生梯度消失。梯度爆炸和梯度消失都会导致神经网络无法有效地进行学习。为解决梯度爆炸和梯度消失，人们先后提出了梯度剪切、采用 ReLU 函数作为非线性单元、批归一化和残差结构等解决方案。其中梯度剪切的方案是通过对传播过程中的梯度设置相应的阈值，将其限制到阈值范围内；采用 ReLU 函数作为非线性单元则是因为 ReLU 函数克服了非线性单元在输入值过大或过小时梯度接近于 0 而容易造成梯度消失的缺点，保证了各层网络的梯度更新速度一致。

神经网络的规模日益增大和任务复杂度的提高使过拟合和欠拟合逐渐成为人们关注的问题。过拟合是指深度神经网络在训练集上的表现很好，但在测试集上的表现较差的现象，也即深度神经网络的泛化能力较差。过拟合通常发生在网络模型很大但训练数据有限的情况下。为解决过拟合的问题，人们提出了数据增广、正则化、神经元随机失活（又称 Dropout）等方法，这些方法的合理运用能有效减少深度神经网络过拟合的发生。

与过拟合相对应，欠拟合则是指深度神经网络在训练集和测试集上的表现都不好的现象。发生欠拟合通常是因为所使用的神经网络模型较为简单，不能很好地学习到训练集中数据的特征，从而训练好的模型不具备较好的预测能力。结合过拟合和欠拟合，能看到神经网络模型的复杂度选择很重要，需要根据不同的任务合理选择神经网络的结构，避免过拟合或欠拟合的发生。

2.2 强化学习基础

强化学习是一种新兴的人工智能算法，在强化学习当中，智能体能像人类一样通过与环境间的交互不断对自身的策略进行调整，最终学习到最优的策略，从而能在某一任务当中获得较好的表现。

2.2.1 强化学习与马尔科夫决策过程

强化学习问题通常能描述为一个马尔科夫决策过程（Morkov Decision Process, 简称 MDP）。如图 2-3 所示，马尔科夫决策过程由智能体、环境、状态、动作和奖励五个部分组成，智能体即学得的策略，环境即除了智能体以外的一切事物的集合。在一个完整的马尔科夫过程中，当前环境处于某个状态，智能体对环境实施一个动作，环境则根据状态转移概率转移到下一个状态，并给予智能体一个奖励。马尔科夫决策过程有一个重要的特点——马尔科夫性，也即当前时刻的状态只与上一时刻的状态以及上一时刻采取的动作有关，而与之前的状态无关。在一个强化学习任务当中，智能体会不断与环境进行交互，直到环境给出一个终止信号，而这一整个过程称为一个回合，而在这个过程当中产生的样本又被称为一条样本轨迹，强化学习算法的目标则是最大化整个回合中的累积奖励。

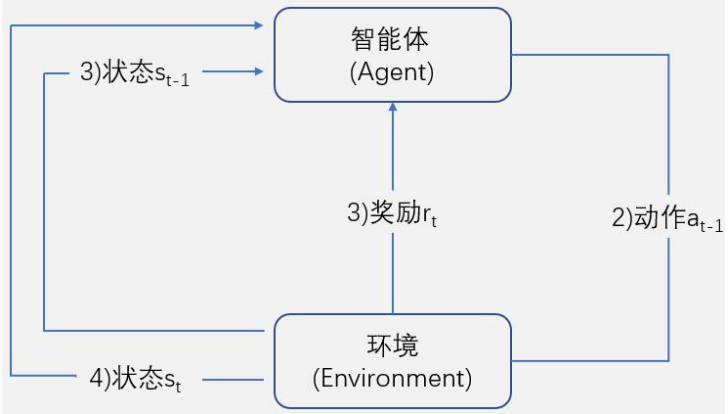

图 2-3 马尔科夫决策过程

由于每个样本轨迹的长短、经历的状态都可能完全不相同，直接对累计奖励进行估算存在估算值变化较大的问题，因此在强化学习领域中，通常使用 V 函数 V(s)表示当前状态 s 的所有后继状态获得的累积奖励值的期望，而使用 Q(s, a)表示在当前状态 s 下采取动作 a，所有可能的后继状态获得的累计奖励值的期望。相较于直接对累积奖励进行估算，对累计奖励值的期望进行估算能使估算结果更为稳定，更有利于强化学习算法的收敛。因此，在很多强化学习算法当中，最大化的目标通常是累计奖励的期望值。

根据样本获取方式的不同，能将强化学习算法分为同策略（on-policy）和离策略（off-policy）两种。在同策略算法当中，获取样本的策略和更新的策略是同一个策略，典型代表为 SARSA 算法；在离策略算法当中，获取样本的策略和更新的策略不是同一个策略，典型代表为 Q-learning 算法。同策略算法训练过程较为稳定，但是每次进行策略更新之前都需要等待整个回合结束，因此训练过程效率较低；离策略更新算法能充分利用先前的样本，数据利用率较高，但训练过程很容易不稳定。

此外，根据状态转移概率是否已知，能将强化学习算法分为模型已知（Model-Based）的算法和模型未知（Model-Free）的算法。模型已知的算法主要求解方式是动态规划算法；而模型未知的算法则有蒙特卡罗方法、时间差分方法以及演员-评论员方法等。

2.2.2 深度强化学习

将深度学习方法跟强化学习算法相结合，便得到了深度强化学习。传统的强化学习算法虽然有效，但是也存在难以找到合适的状态表达方法、容易陷入维数灾难等问题[14]，而深度学习的强项在于特征提取，因此将深度学习融合到强化学习当中能很好地解决这些问题。

将深度学习融入到强化学习算法中有许多方法，大致能将这些方法划分为三类——基于值函数的方法、基于策略近似的方法以及基于其他结构的方法[14],其中基于值函数的方法的思路是使用深度神经网络对值函数进行拟合；基于策略近似的方法的思路是使用深度神经网络对策略进行拟合；基于其他结构的方法则有多种形式，例如一些采用异步思想或分布式思想的算法等。

深度学习的引入极大地提高了强化学习算法的表现，DQN 算法[16]就是其中典型的代表。该算法使用深度神经网络对值函数进行近似，并且采用经验回放机制对之前采样的数据进行处理和存储，极大降低了数据间的相关性，最终的实验结果表明该算法在 Atari 游戏中达到了超越人类的表现[17]。

DQN 算法虽然能很好地解决离散动作空间的问题，但是无法解决连续动作空间下的问题。而 DDPG 算法[20]的提出则为连续动作空间下的问题求解提供了参考的思路。DDPG 算法对值函数以及策略均使用深度神经网络近似，采用了演员-评论员的框架，沿用了 DQN 算法中的目标网络以及经验回放机制，在一些连续动作空间下的任务取得了较好的结果。

AC 框架是强化学习算法的另一个重要内容。在 AC 框架中，通常设立一个评论家网络（Critic）来对 Q 值进行计算，而另外设立一个演员网络（Actor）负责选取动作。在算法流程当中，Critic 结合当前时刻的状态，对 Actor 选取的动作输出一个相应的 Q 值，而这个 Q 值则反馈到 Actor，供 Actor 对自身的动作选择策略进行优化。

然而使用 Q 值作为累积奖励值的估算会有较大的方差。为减小方差，人们设计了优势函数，也即在 Q 值的基础上减去 V 值：

$$A^{\pi}(\mathrm{s}_t, a_t) = Q^{\pi}(s_t, a_t) - V^{\pi}(s_t) \tag{2-1}$$

由于 V 值表示的是当前状态下累计奖励的平均值，因此结果的正负表明了在当前状态下选取该动作的好坏。但是如果将优势函数应用到 AC 框架下会导致

AC 框架需要新增一个 V 值网络。由于同时对 V 值和 Q 值进行估计会导致不稳定，又为减少网络数目，在 AC 框架中通常只使用一个 Critic 网络来估算 V 值，而 Q 值则能通过 V 值与即时奖励值估算得到：

$$Q^{\pi}(s_t, a_t) = r_t + V^{\pi}(s_{t+1}) \tag{2-2}$$

因此，优势函数又能表示为：

$$A^{\pi}(s_t, a_t) = r_t + V^{\pi}(s_{t+1}) - V^{\pi}(s_t) \tag{2-3}$$

这种在 AC 框架下使用优势函数评判动作好坏的算法被称为优势演员—评论家算法（Advantage Actor-Critic，简称 A2C）。

异步优势演员-评论员算法（Asynchronous Advantage Actor-Critic，简称 A3C）[55]在 A2C 算法的基础上，采用异步的思想，在 CPU 上使用多线程进行训练。在 A3C 算法当中，有一个全局网络和若干个局部网络，这些网络的结构完全相同。每个线程上的智能体在各自的环境中进行数据采集，并利用采集到的数据计算得到网络更新得到的梯度值，最终将这些梯度回传到全局网络，供全局网络进行更新；而全局网络更新后，会将网络参数与各个局部网络进行同步。训练时，梯度回传和参数同步的流程不断重复，直到全局网络收敛。A3C 算法在降低数据间相关性的同时，还极大提高了训练效率。A3C 算法的流程如图 2-4 所示。

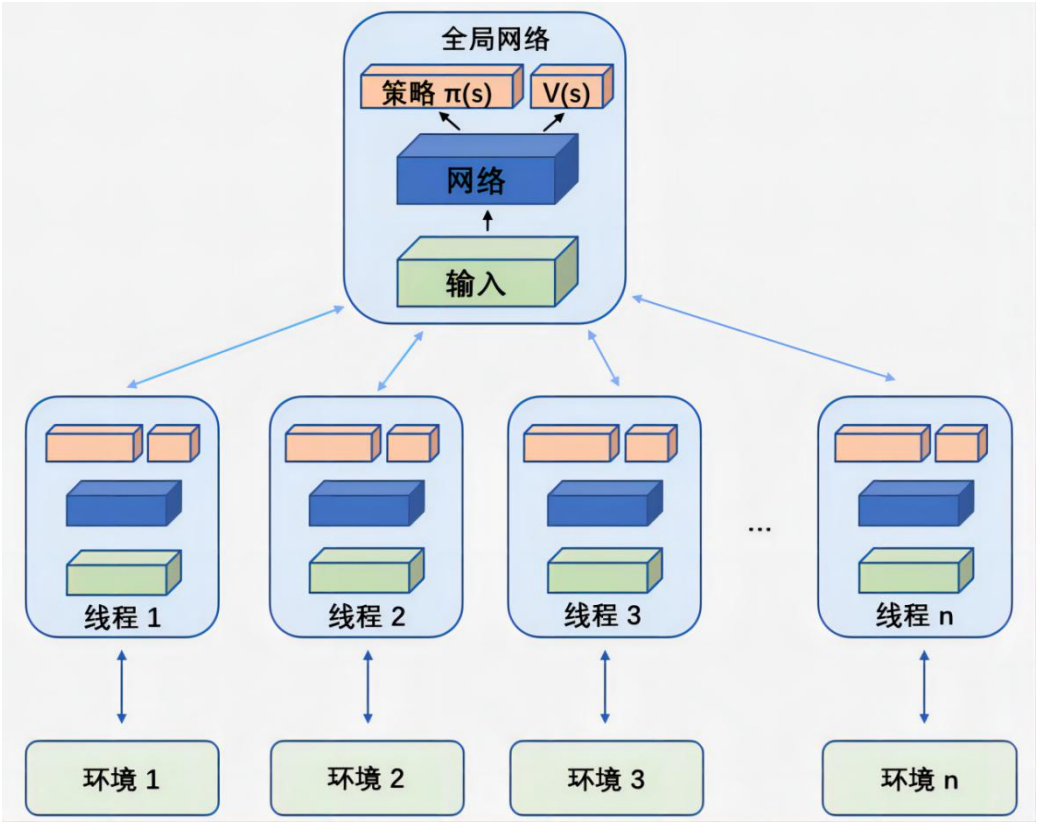

图 2-4 A3C 算法流程

为进一步加快 A3C 算法的训练效率，并且能充分利用 GPU 设备，Nvidia 推出了 GPU 版本的 A3C 算法 GA3C[56]，这使深度强化学习算法的实用性进一步增强。

图 2-5 是深度强化学习算法的分类图，从中能看出，深度学习的加入使强化学习算法的种类得到了大大的扩充。

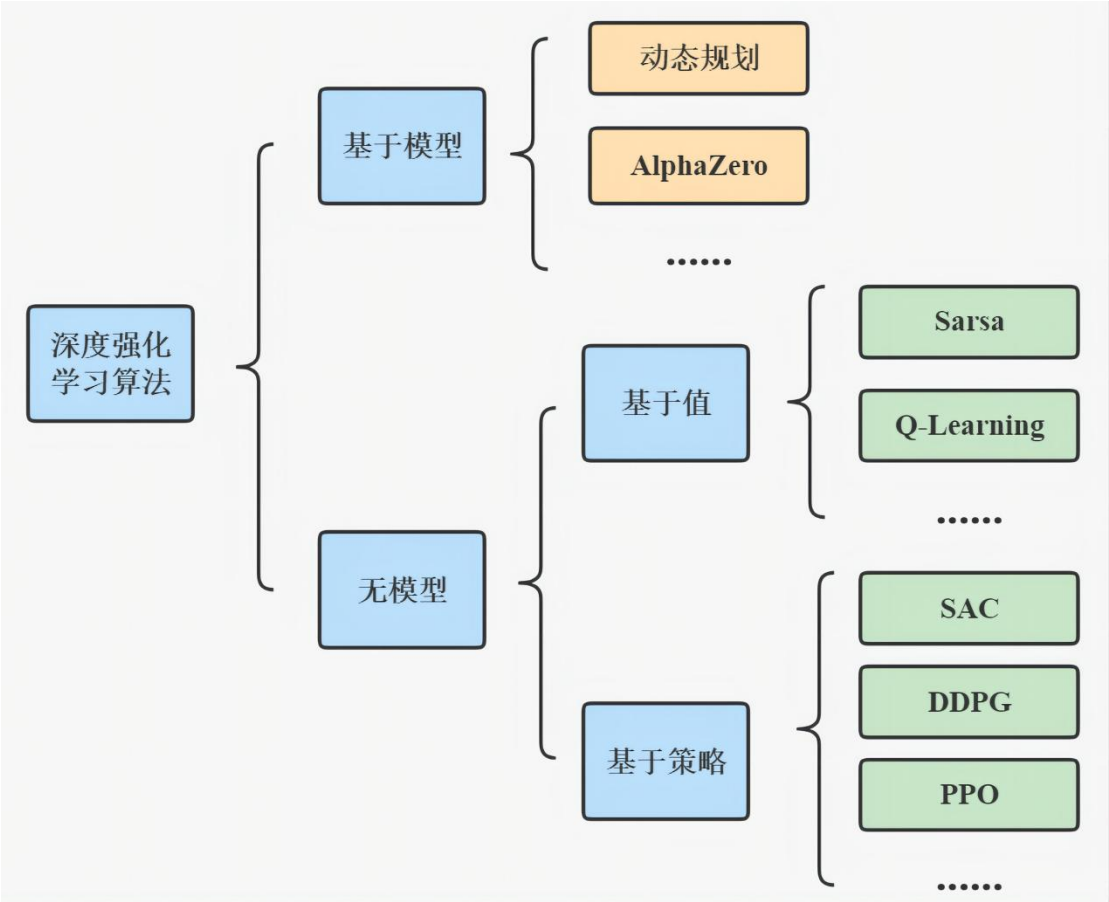

图 2-5 深度强化学习算法分类图

2.3 本章小结

本章主要对深度学习、强化学习以及两者的结合——深度强化学习的理论基础进行了归纳、总结和介绍，最后对本文工作中用到的主要深度强化学习算法，即 A3C 算法进行了较为详细的介绍。

第 3 章　基于深度图和目标检测的目标导航方法

3.1 问题阐述与分析

图 3-1 展示了 Y.Zhu 等人设计的网络结构[41]，该网络结构将当前时刻的 RGB 观测图像和机器人在目标点处以第一人称视角获取的 RGB 图像（下称目标点图像）同时经过 ResNet-50 模型[42]进行预处理后，提取得到的特征图各自连接到一个全连接层，最后两个全连接层的输出将融合到一个大的全连接层中（该结构被称为孪生网络）。孪生网络的输出作为强化学习模块的状态输入，供强化学习模块使用。与只输入当前时刻状态的深度强化学习方式[16][53]不同，由于其模型输入中包含了目标点图像，同时在训练时采用了类似于 A3C 算法的多线程训练方法，同时使用多个目标点对模型进行训练，因此该模型最终具有较强的跨目标泛化能力。但实验同时也表明，该模型在在跨场景泛化能力上还有很大的改进空间。

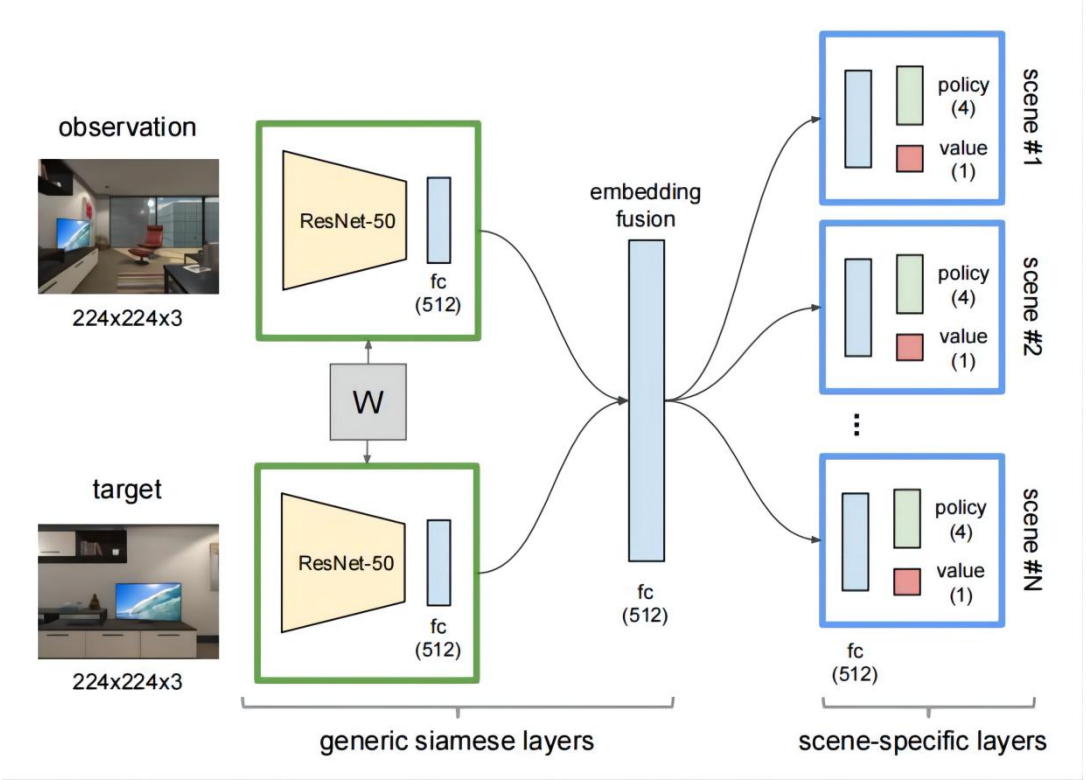

图 3-1 Y.Zhu 等人设计的网络结构[41]

经过分析，本文认为造成模型跨场景泛化能力差的主要原因有两点：

1）模型以原始的 RGB 图像直接作为输入，虽然能较为方便地实现端到端的训练与控制，但原始的 RGB 图像中包含过多场景特有的信息，而这些信息会在训练时会隐式地编码记录到网络参数中，从而导致模型在新场景中表现不佳；

2）模型使用 ResNet-50 网络对输入进行预处理，但是 ResNet-50 网络是在

ImageNet 数据集上进行训练的，因此网络提取的特征可能会忽略与导航任务相关的信息，造成模型泛化能力较弱。

对于问题 1），A.Mousavian 等人的工作[43]表明，不同的视觉表征方式对于目标导航模型的泛化能力有着较大的影响，相较于使用 RGB 原始图像作为视觉表征方式的模型，使用语义分割结果、目标检测等高级语义信息作为视觉表征方式的模型在泛化能力方面的表现更好。这是因为语义分割和目标检测等高级语义信息中包含的场景特定信息较少，将高级语义信息作为模型输入，模型能忽略掉场景特有的信息，而专注于导航任务，因此最终能获得更好的泛化能力。

联想到人类自身和动物的导航方式，本工作认为成功完成导航任务的关键有两点——寻找目标和避障。寻找目标即能在导航过程中分析出当前时刻观测到的物体种类与位置，并将其与目标物体进行对比，从而确定自身下一步行动的能力；避障则是根据当前时刻自身的位置以及分析自身与周边障碍物的距离从而选择合适的动作避开障碍物的能力。

若要将这些关键点应用到机器人目标导航任务中时，智能体在导航过程应当能中识别出当前时刻观测中的物体种类及其位置，能测量出当前时刻自身与周围障碍物间的距离，并将这些感知结果与目标物体的信息进行综合分析，选择出当前时刻的最佳动作。

对此，结合 Y.Zhu 等人的工作[41]以及 A.Mousavian 等人的工作[43]，本文提出了一种结合目标检测结果和深度图的深度强化学习视觉目标导航模型，目标是以最少的步骤数导航至以图像形式指定的目标点。如图 3-2 所示，该模型同时将当前时刻的深度图和 RGB 观测图像以及目标点图像作为输入，模型对输入信息进行处理后输出当前时刻应该采取的动作。

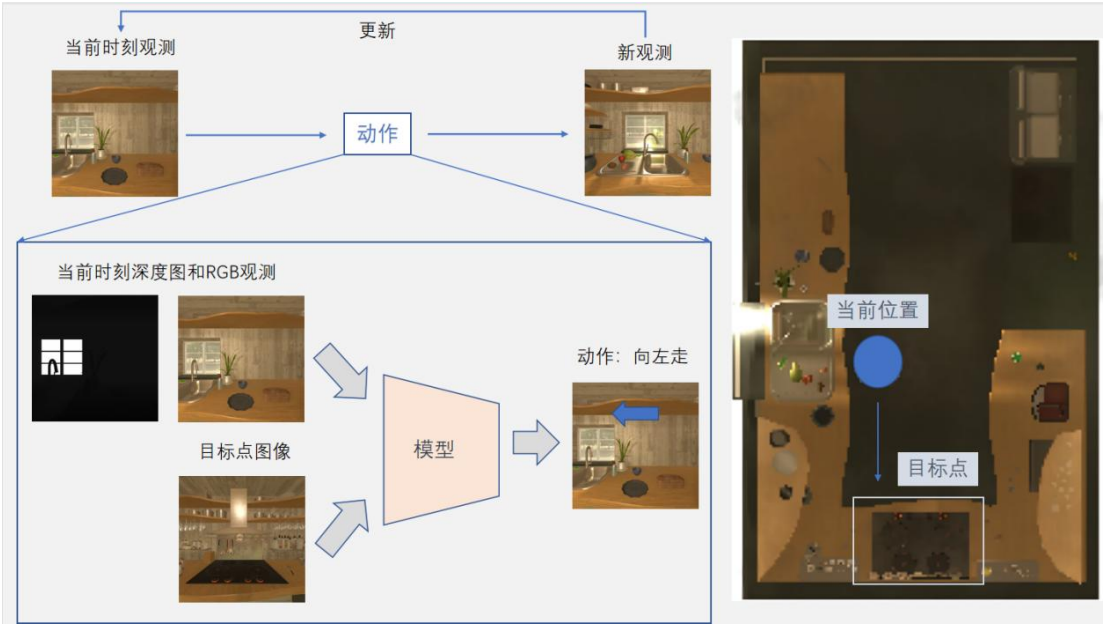

图 3-2 模型的基本工作流程

3.3 基于目标检测和深度图的视觉目标导航方法

3.2.1 状态表示方式

为使智能体具有寻找目标物体的能力，本工作使用目标检测模块对观测与目标的 RGB 图像进行了预处理。目前主流的目标检测模块有 R-CNN 系列[58-60]、YOLO 系列[61-64]和 SSD 系列[65-70]等。目标检测模块能识别出图像中物体的种类，并使用标注框的形式标注出其在图像中的位置。在本工作中，智能体在导航过程中将当前时刻对环境的 RGB 观测图像和目标点图像同时输入到目标检测模块中，经由模块处理后输出观测图像中物体的种类与位置。

为能让智能体学习到如何根据当前时刻的状态进行避障，本文在状态的表示中引入了深度图。深度图是一种由深度相机等传感器获取得到的图像，图像上的每个像素数值的大小表示场景中该点与相机间的距离。由于深度图包含了相机前方各个位置的空间距离信息，因此十分适合用于作为避障的参考。此外，与环境的原始图像相比，深度图中包含的与环境相关的细节信息较少，对于模型跨场景导航泛化能力的提高也能有一定的帮助。

在 A.Mousavian 等人的工作[43]中，假设目标检测器能检测 N 种物体，为将目标检测的结果与原始 RGB 三通道图像相结合，原工作在原始 RGB 三通道图像的基础上增添了 N 个新通道，其中第 i 个新通道对应一种目标检测器能检测的第 i 个物体类别，而每一个物体类别对应的标注框信息则会被绘制到对应的新通道中对应的位置（对于没有检测到的物体类别，其对应的通道则保留原有数值），因此如图 3-3 所示，最终输入到模型中的张量大小变为 $H \times W \times (3+N)$。

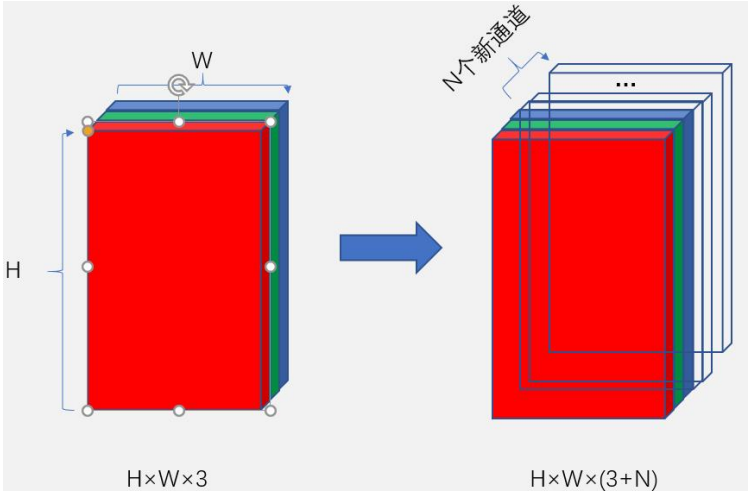

图 3-3 A.Mousavian 等人的工作[43]结合目标检测结果的方式

该方式虽然能在不破坏原始 RGB 图像信息的情况下将目标检测结果添加到图像中，但该方法只适用于物体类别较少且物体类别数量固定的情况下（原工作

中物体类别数量仅有 3 种），并且包含大量的冗余信息，造成计算资源的浪费。

为避免上述问题，本工作选择将目标检测的结果以白色标注框的形式绘制到当前时刻的深度图当中作为状态的表示方式，由于白色标注框会与深度图中灰黑色的像素形成鲜明对比，产生了明显的梯度变化，因此在对状态表示图利用卷积运算进行特征提取时，也很容易将这一信息提取出来。这一状态表示方式能在极少地破坏原图信息的同时将目标物体的信息表示出来，并且不会因为目标检测器能检测的物体数量较多而带来过多的冗余信息。

此外，与 Y.Zhu 等人的工作[41]不同，由于本工作使用目标检测器对输入 RGB 图像预处理，多次处理相同的 RGB 图像实质上造成了冗余的预处理操作，进而导致预处理效率的降低，因此本工作并没有将目标点图像每次都作为模型的输入。同时，为避免模型在训练时缺乏目标点的信息而导致跨目标泛化能力弱，本工作将目标点图像的目标检测结果信息融入到当前时刻状态表示以及奖励函数中

（1）融入到当前时刻状态表示

设目标点图像的目标检测结果为集合 U，而当前时刻 RGB 观测图像的目标检测结果为集合 V，则以白色标注框的形式绘制到当前时刻深度图中构成当前时刻状态表示的目标检测结果为 U⋂V，也即当前时刻机器人观测到的集合 U 中的目标物体。通过该种方式，本工作希望智能体能在导航过程中借助目标点图像中检测到的目标物体进行定位；

（2）融入到奖励函数

该部分内容将在 3.2.2 一节“奖励函数设计”中详细介绍。

图 3-4 展示了获得某一时刻状态表示的过程。能看到，这种表示方法综合了对目标物体位置以及当前时刻前方场景深度信息，因此完全有可能让智能体实现在尽可能快地向目标靠近的同时也能实现避障。

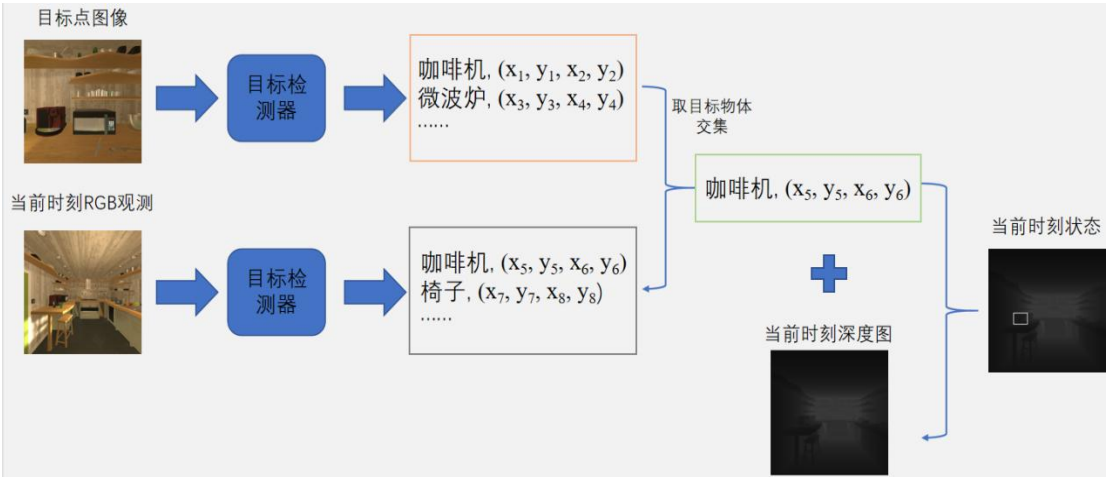

图 3-4 结合目标检测结果以及深度图的状态表示

3.2.2 任务设计

（1）动作空间设计

为降低实验难度，任务中的动作空间只包含几个简单的动作——向前移动 0.5 米、向后移动 0.5 米、向左旋转 90°、向右旋转 90°和停止。离散的动作空间将 AI2THOR 中的场景划分为一个网格世界，每个场景仅有有限个可到达的位置。图 3-5 展示了 AI2THOR 中的 FloorPlan3 场景的整体布局以及可到达的位置（图中蓝点）在场景中的分布：

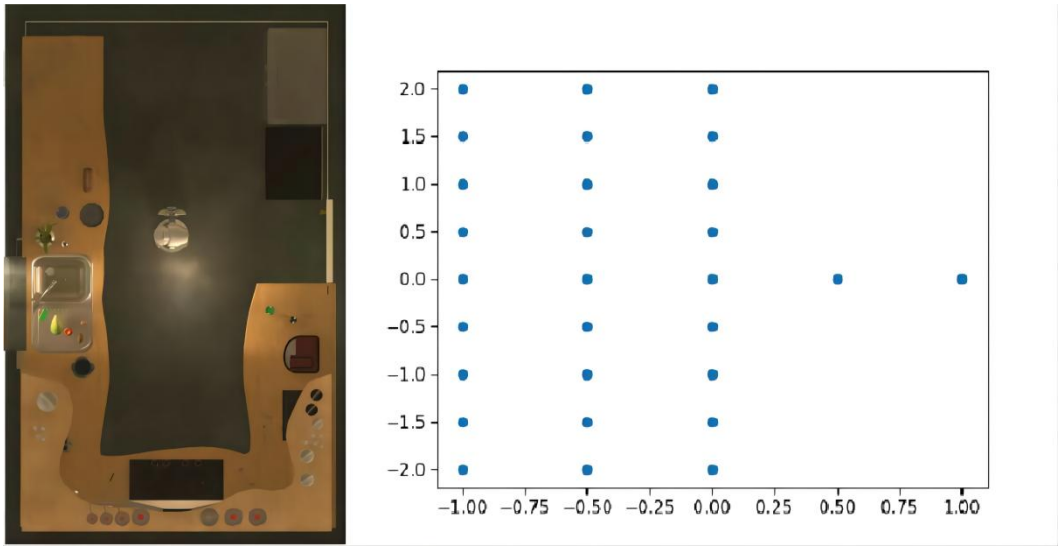

图 3-5 FloorPlan3 场景布局（左）及场景中可到达位置分布（右）

（2）观测与目标

智能体的观测和目标均包含智能体第一视角下获取的 RGB 图像，除此之外，智能体第一视角下获取的深度图也作为观测输入之一，所有的观测图像和目标图像的大小均设置为 224×224。使用 RGB 图像作为目标的表现形式能让目标点的设置和更换变得更为方便。在任务当中，智能体会被提供一张目标点的 RGB 图像，并被要求自主导航到获取目标点 RGB 图像的位置附近；

（3）奖励函数设计

奖励函数作为智能体策略学习的一个重要引导方式，成为强化学习问题中的研究重点之一。为引导智能体在尽可能快地接近目标的同时也实现避障，本文提出了一种新的复合奖励函数设计方案，该方案主要由以下三部分组成：

1）目标检测奖励

为引导智能体尽快寻找到目标，本工作在奖励函数中添加了物体目标检测框大小这一因素。若智能体没有检测到目标物体，则会获得一个较小的惩罚-0.04；若智能体能检测到目标物体，会得到一个固定奖励 0.02 和一个可变奖励，该可变奖励大小与各个目标物体的检测框面积大小成正比。此外，为消除创建环境时设置观测图像大小不同为奖励计算带来的影响，计算目标检测奖励时使用的是归

一化之后的检测框大小。

本工作中的目标检测奖励 R_{objdet} 能用公式表示为：

$$R_{obj\,\det}=\begin{cases}-0.04 & n=0\\ 0.02+\sum_{i=0}^{n}0.02\cdot S_i & n>0\end{cases} \tag{3-1}$$

式中 n 为当前时刻检测到的目标物体个数，S_i 为第 i 个目标物体归一化后的检测框的面积。这种设置奖励的方法能让智能体更关注于目标导航而不是过于关注场景特有的因素，能尽快找到目标物体，并且对体积较大的物体更感兴趣，对于提高智能体的跨场景泛化能力有较大的帮助。

2）碰撞惩罚

为引导智能体学会避障，本工作在智能体发生碰撞时给予一个惩罚-0.04。这会使智能体在导航过程中能尽可能地避开障碍，直接前往目标位置。

碰撞惩罚 $R_{collided}$ 用公式表示为：

$$R_{collided}=\begin{cases}-0.04 & collided\\ 0 & otherwise\end{cases} \tag{3-2}$$

3）导航奖励

由于导航工作最关注的还是导航路径的长短。因此，在智能体没到达目标时，会给予一个较小的惩罚-0.01；当智能体到达目标时，会得到一个较大的奖励 10。

导航奖励 R_{nav} 用公式表示为：

$$R_{nav}=\begin{cases}-0.01 & finished\\ 10 & otherwise\end{cases} \tag{3-3}$$

因此，某时刻的总即时奖励 R 能表示为：

$$R=R_{obj\,\det}+R_{collided}+R_{nav} \tag{3-4}$$

（4）终止状态判断

与 Y.Zhu 等人的工作[41]一致，本工作使用了一种基于位置与朝向信息的终止状态判断的依据。该判据认为，只有智能体当前时刻的位置与朝向与获取目标点图像时的位置与朝向完全一致时才判定为智能体到达了终止状态。这种判据简单、较为严格，适合应用于对导航结果有严格要求的场合。

3.2.3 模型与训练方案

（1）模型

基于深度强化学习的目标导航方法是通过深度强化学习获得一个目标驱动的策略函数 π。本工作设计了一个深度神经网络模型作为函数 π 的逼近器，能把模型用如下公式表示：

$$a \sim \pi(s_t, g \mid \theta) \tag{3-9}$$

其中θ表示模型参数，s_t是当前时刻观测的 RGB 图像，g 是目标点的 RGB 图像，a 则是智能体采取的动作。

模型将当前时刻观测的 RGB 图像和目标点的 RGB 图像作为输入，输入分别经过目标检测模块进行预处理后，会得到两个物体检测列表，这两个列表的交集会被视为智能体当前时刻检测到的目标物体，之后这些目标物体对应的检测框会被绘制到当前时刻的深度图上，以表示这些物体与智能体间的空间关系。最终的绘制结果会作为当前时刻的状态表示，输入到后续的强化学习模块。

除此之外，模型中还包含一个奖励计算模块，其主要工作是根据当前时刻的两个物体检测列表以及对应检测框的大小等信息计算当前时刻的即时奖励。

模型的整体框架如图 3-6 所示。其中当前时刻的 RGB 观测图像与目标点图像同时输入到目标检测模块进行目标物体的检测，检测结果将同时传输到奖励函数计算模块以及智能体，奖励计算模块根据得到输入信息计算当前时刻的奖励，并将计算结果反馈给智能体，智能体则综合输入的各种信息进行下一时刻的决策。

（2）训练方案

在深度强化学习算法选择方面，传统的深度强化学习算法通常只能设置单个目标进行训练，这样会导致训练好的模型不具备良好的跨目标泛化能力。而 A3C 算法[53]是一种采用异步的方式，并行运行多个复制的训练线程，对一个共享的网络进行参数更新的算法，该算法适合同时设置多个目标进行训练，因此本工作参考了 Y.Zhu 等人工作中的方案[41]，采用了一种类似于 A3C 算法的训练方案对模型进行训练。与 A3C 算法中各个训练线程均运行同一个游戏的拷贝不同，该训练方案中各个线程都设置了不一样的导航场景或导航目标，因此各个训练线程回传到全局网络的梯度能相互平衡，使最终训练出来的模型具有较好的跨目标泛化能力。

在目标检测模块方面，本工作选用了体积小、推理速度快、性能适中的 YOLOv5s 作为目标检测模块；在训练时，为能直接使用 OpenCV 中的检测框绘制工具且保持图像信息不变，本工作将单通道深度图复制三份，合成一个三通道的“伪深度图”；强化学习部分则先由一个共享的小型卷积网络对状态输入进行特征提取，而 Actor 和 Critic 均由一个具有相同大小的输入层和隐藏层的全连接神经网络组成，特征提取部分所得到的特征图则作为 Actor 和 Critic 的输入。最后，本工作采用了一个共享参数的 Adam 优化器分别对 Actor 和 Critic 网络进行训练，初始学习率设置为7×10^{-3}。

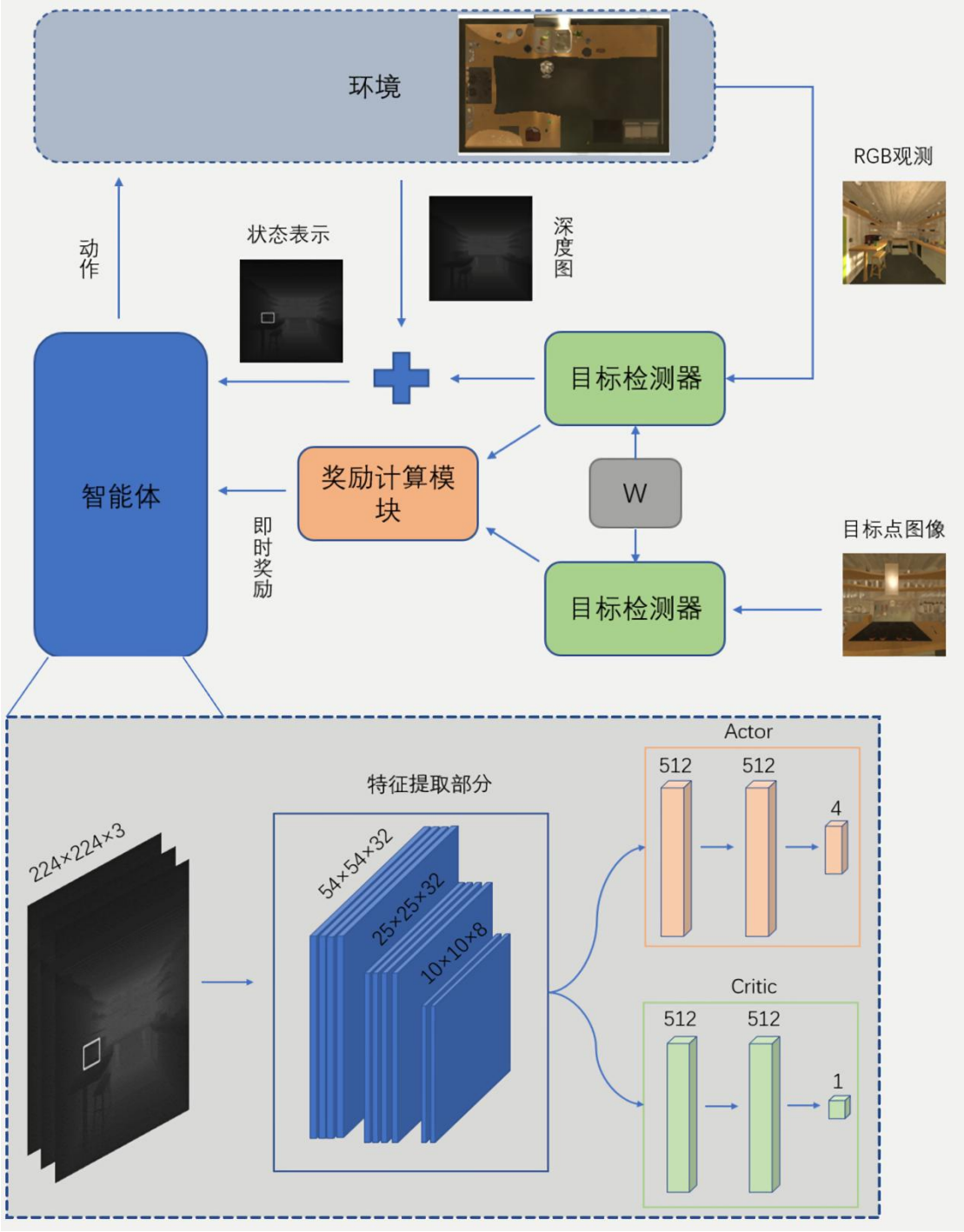

图 3-6 模型的整体框架

3.3 本章小结

本章主要对本文的主要工作进行了详细介绍。首先分析了现有视觉目标导航方法存在的问题，然后在模型设计、状态表示方法、奖励函数以及训练方案等各个方面对本文所提出的视觉目标导航模型进行了详细介绍。

第 4 章　实验结果与分析

4.1 AI2THOR 仿真环境

在本节中先对实验所使用的 AI2THOR 仿真环境进行简单介绍，最后介绍离线 AI2THOR 数据集的相关内容。

4.1.1 AI2THOR 简介

在基于深度强化学习的目标导航领域，常常需要使用到仿真平台对智能体进行训练。在训练过程中，智能体通过动作与仿真平台进行交互，仿真平台根据智能体的动作进行环境的改变，并返回一定的反馈；而智能体则根据仿真平台的反馈对自身策略进行调整，进而达到从交互中学习的目的。常用的仿真平台有 AI2THOR[27-29]、Habitat[30-31]和 iGibson[71]等。

AI2-THOR 是美国 Allen 人工智能研究所基于 Unity3D 游戏引擎开发的一个室内环境仿真平台，该平台由 iTHOR[27]、RoboTHOR[28]和 ManipulaTHOR[29]三种不同类型的场景组成，其中 iTHOR 是一个室内环境仿真场景，RoboTHOR 则是一个同时包含真实与模拟的室内场景，ManipulaTHOR 则是一个针对机械臂开发的仿真场景，本工作主要基于 iTHOR 完成。目前 iTHOR 共提供了四种不同类型的场景，分别是厨房、起居室、卧室和浴室，每种场景又有三十种不同的房间，每种房间在大小、装饰、物品布局等方面均不相同。

AI2THOR 仿真平台具有较强的交互性，除了前进、后退、旋转等普通的动作之外，智能体还能拾起、放下、打开、关闭场景中的一些物体。AI2THOR 中的物体类型多达一百多种，每种物体拥有不同的属性，智能体能根据物体属性的不同施加不同的动作改变其状态。此外，为方便用户使用，AI2THOR 还提供了丰富的 Python 编程接口，包括场景的切换与初始化、智能体与环境的互动、对智能体观测信息的提取等。

4.1.2 离线 AI2THOR 数据集

即使是将 AI2THOR 仿真环境放到 GeForce RTX 3060 GPU 上运行，当渲染质量为中等、渲染图像大小为 224×224 时，实时渲染的帧率也仅有 3.6fps，并且渲染得到的深度图也会因为渲染质量差而可能出现残缺的现象，因此这完全无法满足本工作的模型训练的需求。为了加快模型与仿真环境之间的交互速度，本工作将 AI2THOR 中的所有仿真场景制作成一个数据集，由于与直接使用 AI2THOR 仿真平台相比，数据集省去了 Unity3D 实时渲染时所需要的 HTTP 请

求等繁杂的通信过程，因此本工作称之为离线 AI2THOR 数据集（Offline AI2THOR Datasets）。

如图 3-5 所示，由于 AI2THOR 仿真场景为室内封闭场景，且对机器人的每一步运动距离和旋转角度均进行了限定，因此其状态空间为一个离散、有限的空间，只需要记录下每一个状态下的观测等信息以及各状态间的转移关系，即可将整个仿真场景制作为一个数据集。

本工作以表 4-1 中的仿真环境设置参数运行每个 AI2THOR 仿真场景。由于机器人每次旋转的角度都固定为 90°，因此每个位置下共有 4 个状态。

表 4-1 AI2THOR 仿真环境参数设置

仿真环境参数列表	
图像宽高	480×640
机器人每次运动的固定距离	0.5m
机器人每次旋转的固定角度	90°
相机视场角	90°
机器人最大可视距离	2.5m
动作列表	前进、后退、向左旋转、向右旋转

在离线 AI2THOR 数据集中，为方便基本的导航交互需要，本工作将场景中每一个状态下的 RGB 观测图像、深度图以及位置和朝向信息记录到 hdf 文件中，并通过机器人与环境间的交互和各状态间的换算关系获得各状态间的转移关系；为方便进行成功率权重的最短路径长度（即 SPL）[74][75]等指标的计算，数据集提供了各个状态间的最短距离；为方便进行导航路径的可视化，数据集为每个状态提供了相机高度为 2.0m 的仿真场景俯视图。最后，考虑到对于 AI2THOR 仿真环境参数设置需求不同，本工作提供了制作数据集的 Python 脚本。

表 4-2 清晰地展示了使用离线 AI2THOR 数据集与使用 AI2THOR 仿真环境实时渲染的区别。可以看到，使用离线 AI2THOR 数据集不仅能大大加快与仿真环境的交互速度，还能实现 AI2THOR 仿真环境的跨平台应用。

表 4-2 离线 AI2THOR 数据集与 AI2THOR 仿真环境实时渲染比较

	速度	跨平台
离线 AI2THOR 数据集	4.39 s/百万帧	Windows、Linux 等
AI2THOR 实时渲染	77.16 h/百万帧	仅 Linux

4.2 实验结果与分析

本工作的主要目标是使智能体找到当前位置与目标点间的最短轨迹。在这一部分里，先展开模型与随机游走模型、最短路径以及 Y.Zhu 等人的工作（以下称 Baseline）[41]在跨目标泛化能力和跨场景泛化能力这两方面的比较；最后展开一个补充实验，测试不同性能的目标检测模块对模型导航性能的影响。

4.1.1 目标导航实验与结果分析

本工作使用 PyTorch 框架[72]实现了所提出的模型，并将模型放到 GeForce RTX 3060 GPU 上进行训练，训练的方案按照 3.2.3 节中的“训练方案”执行。为避免在训练过程中始终只对一个目标点进行训练，本工作随机选取了 20 个 AI2THOR 中的场景，在每个场景中，均随机选取了 5 个不同的目标进行训练。

本工作一共使用了 100 个独立的训练线程对模型进行训练，每个训练线程的目标设定为 100 个训练目标中的一个。在训练时，每个训练线程独立地采集经验数据并计算梯度，得到的梯度将用于对全局网络进行更新。本工作对模型进行了 2×10^7 步的训练，其中平均每训练10^6步大约需要 1.2 个小时。

最终得到的模型(Ours)将与以下几种模型在导航表现上进行比较：

（1）随机游走模型(Random Walk)

随机游走模型是最简单的导航模型，智能体在环境中随机采取动作空间中的动作，直至到达目标点；

（2）最短路径(Shortest Path)

最短路径为通过目标点的位置以及当前智能体的位置，由 Dijkstra 算法[76]推算出在网格世界中到达目标点的最短动作序列；

（3）基线(Baseline)

Baseline 即 Y.Zhu 等人的工作[41]，在本实验中，为保证实验条件的一致，本工作采用了与本工作相同的训练方案对 Baseline 同样训练了 4×10^7 步。

最终的表现将由平均轨迹长度（也即多条导航轨迹的平均步数长度）这一指标进行评估，每当智能体抵达目标点，或其走了5×10^3步之后，就认为一个回合结束。本工作将训练时所采用的 20 个场景用于测试，其中每个场景随机选取了 5 个目标，其中每个模型都在每个目标上测试了 10 个回合。对于每个模型，在每个回合开始之前，都将它们随机初始化到场景中的一个位置，最终该模型在该目标中的表现以 10 个测试回合的平均轨迹长度表示。实验结果如表 4-3 所示。

表 4-3 模型导航表现

智能体	平均轨迹长度
随机游走模型	1212.4
最短路径	7.9
Baseline	845.4
Ours	**616**

本工作所提出的模型的测试表现明显优于 Baseline 和随机游走模型，但相较于最短路径模型仍有较大的提升空间。图 4-1 是本工作中模型的学习曲线与 Baseline 的学习曲线对比。相较于 Baseline，本工作的模型具有更高的数据效率。

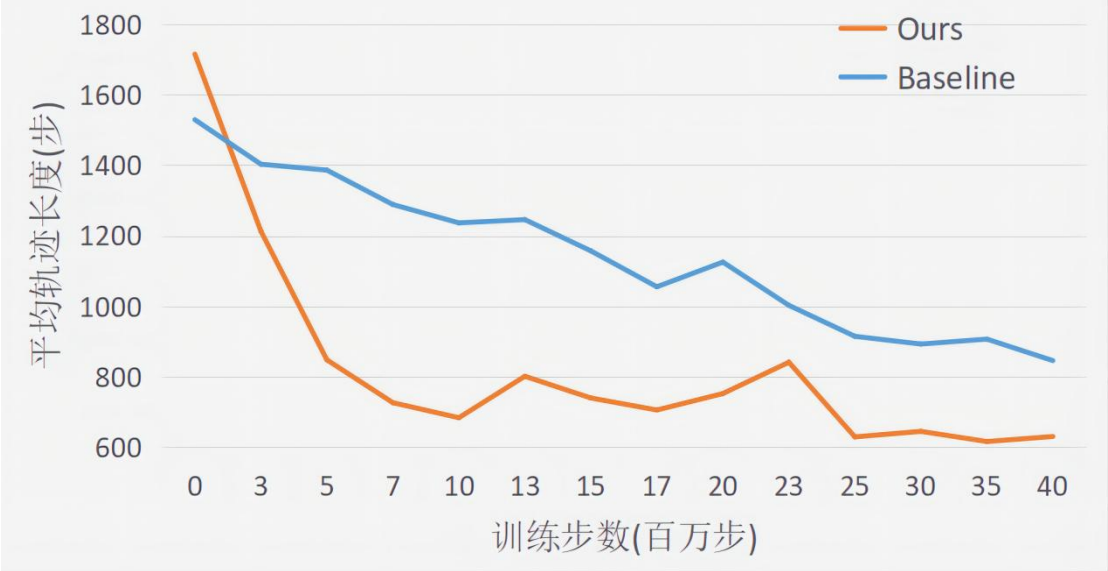

图 4-1 训练时的数据效率

4.1.2 跨目标泛化能力实验分析

由于模型输入包含目标点的图像，因此本工作认为模型应该具备较强的跨目标泛化能力。为能更合理地对模型跨目标泛化能力进行测试，本实验在 20 个训练场景当中选取了一些在训练过程中没有用到的目标点进行测试。对每个目标点，均进行了 100 个回合的测试。所有参与本实验的模型均经过了 2×10^7 步的训练（随机游走模型和最短路径除外）。

本文认为导航模型如果能快速地导航到指定的目标点，那该次导航任务就是成功的，反之则是失败的，因此本实验同时使用成功率（定义为轨迹长度小于 500 步的轨迹占比）和平均轨迹长度两个指标来表征模型的跨目标泛化能力，其中成功率能表示模型是否能足够快速地到达目标点，而平均轨迹长度则能表示模型的整体导航表现是否高效。实验结果如表 4-4 所示。

表 4-4 跨目标泛化能力实验结果

智能体	平均轨迹长度	成功率
随机游走模型	1249.7	44.4%
最短路径	6.7	100%
Baseline	1067.2	47.6%
Ours	**751.9**	**58.2%**

从实验结果中可以看到，相较于 Baseline，本工作所提出的模型同样具有较强的跨目标泛化能力。这表明本工作所使用的状态表示方法同样有利于模型的跨目标泛化能力的提升。

4.1.3 跨场景泛化能力实验分析

提升深度强化学习视觉目标导航模型的跨场景泛化能力是本工作的主要目的。为对模型的跨场景泛化能力进行测试，实验随机选用了 AI2THOR 中没有在训练时用到的 20 个场景。对于每个场景，均随机选用了 5 个目标，并对每个目标进行 10 个回合的测试。所有参与本实验的模型均经过了 2×10^7 步的训练（随机游走模型和最短路径除外）。得到的实验数据如表 4-5 所示。

表 4-5 跨场景泛化能力实验结果

智能体	平均轨迹长度	成功率
随机游走模型	1128.5	42.4%
最短路径	6.6	100%
Baseline	1472.3	37.4%
Ours	**576.2**	**68.6%**

从实验结果可以看出，将实验场景更换为训练时未使用的新场景后，Baseline 的导航表现受到了较大的影响，整体导航表现下滑较为严重；而本工作的模型并没有因此受到显著的影响，仍然保持了相对较好的导航表现（甚至略有提升），这也体现出了本模型跨场景泛化能力强的特点。

此外本工作认为，模型的跨场景泛化能力会随着其训练时所用到的场景数增多而增多，并且训练好的模型在新场景中进行微调之后能具有更好的导航表现。为对该思路进行验证，本工作分别训练了 5 个模型（下文分别称为模型 1 到模型 5），每个模型训练使用的场景数量不一样，其中模型 1 到模型 5 分别训练了 1、2、4、8、20 个场景，所有的模型均训练了 2×10^7 步。图 4-2 为这些模型在无微调的情况下的跨场景泛化能力表现。

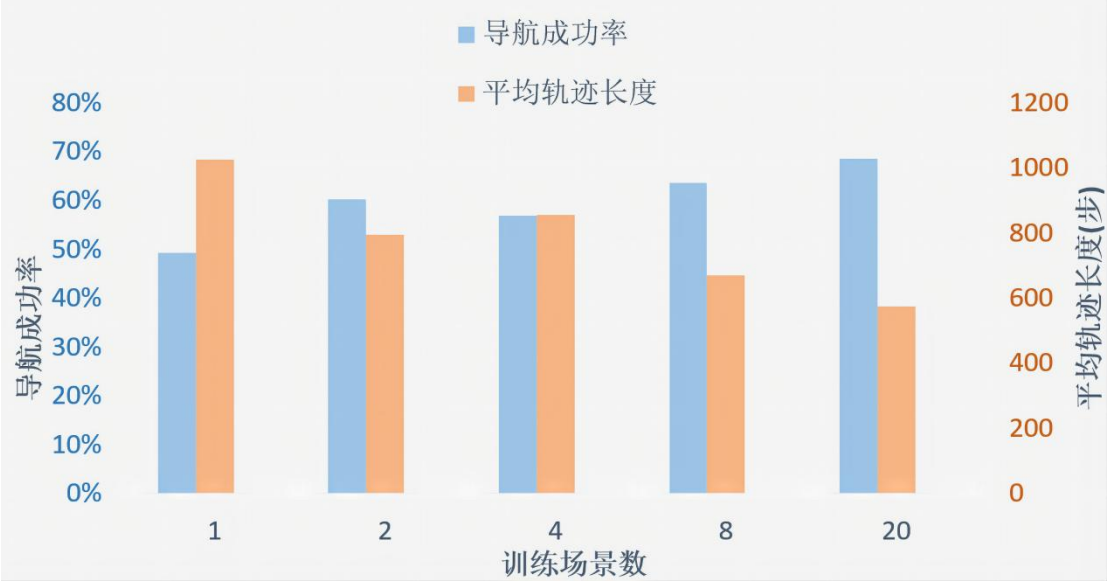

图 4-2 无微调时的跨场景泛化能力实验结果

可见，模型训练时使用的场景越多，其跨场景导航时的平均轨迹长度就会越低，成功率则会越高，也即模型能在新场景中更快地导航到目标点，拥有更强的跨场景泛化能力。

为验证将模型在新场景中进行微调后是否能获得更好的导航表现，下面本工作对这些模型均进行了10^6步到10^7步的微调。对每个模型，冻结其卷积层部分（也即输入预处理部分）的参数，仅在新场景下对 Actor 和 Critic 的全连接神经网络部分进行微调。各模型微调后在新场景中的导航表现分别如图 4-3 和图 4-4 所示。

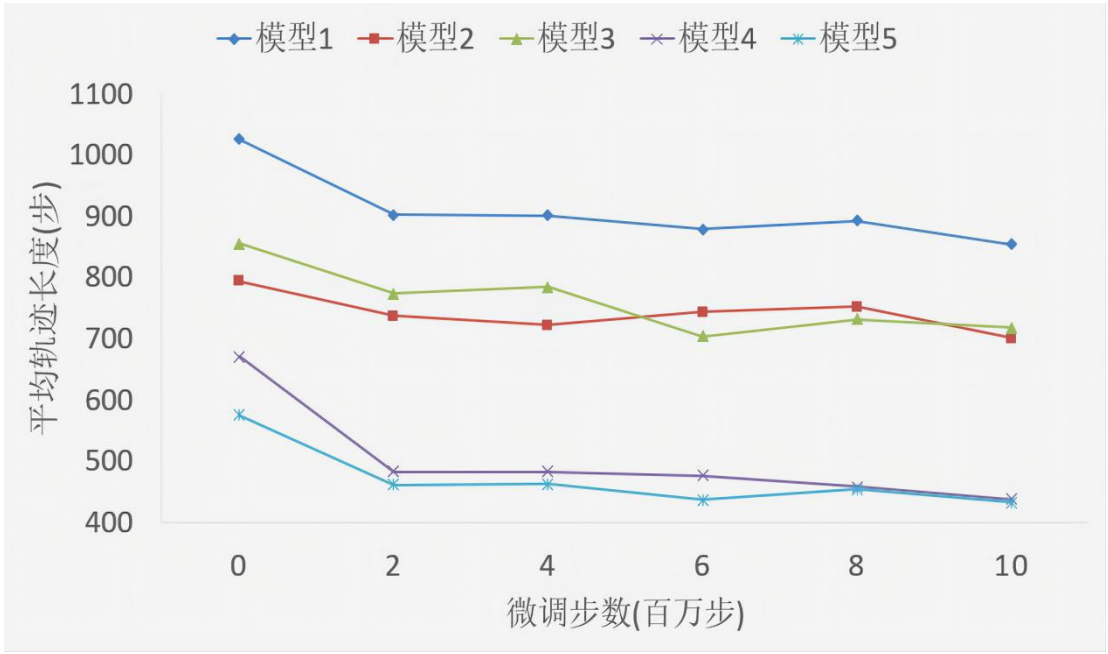

图 4-3 微调后的跨场景泛化能力实验结果（平均轨迹长度）

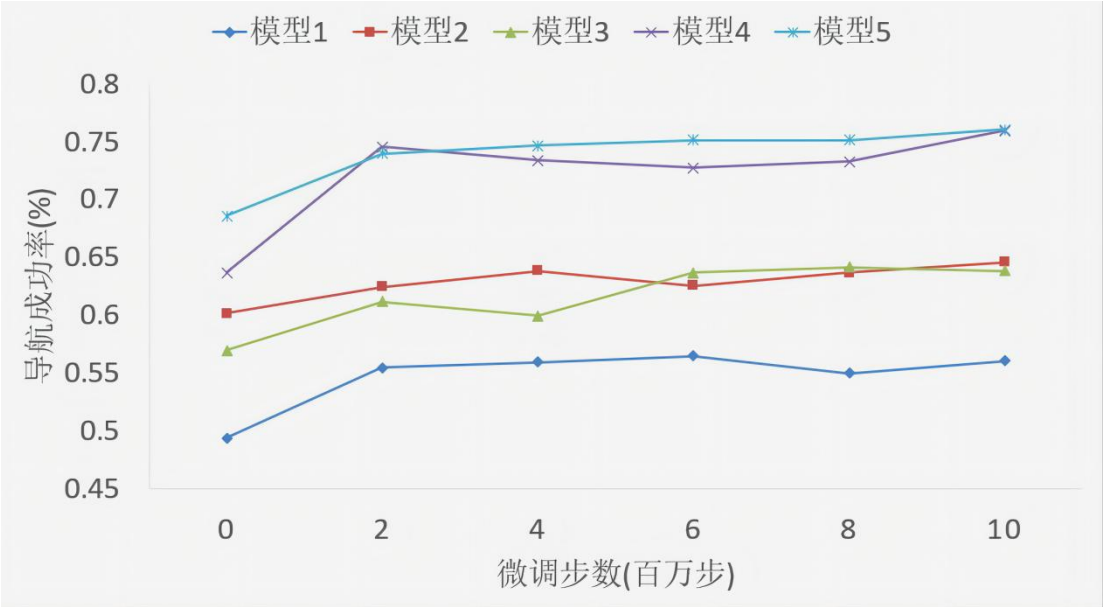

图 4-4 微调后的跨场景泛化能力实验结果（导航成功率）

由实验结果能看出，微调后各模型的平均轨迹长度下降，导航成功率上升，说明各模型的跨场景导航表现得到了提升。

综合以上实验结果，可以得出结论，由于本工作所提出的模型所使用的状态表示所包含的场景特有信息较少，因此模型在新场景中无需调整即可获得较好的导航表现，并且在新场景中对模型进行微调可以使模型的导航表现进一步得到提升，具有较好的跨场景泛化能力。

4.1.4 目标检测模器性能对导航表现的影响

在目标检测领域，衡量目标检测器的指标主要是 mAP:0.5 值（以下简称 mAP 值，该数值指 IoU 阈值为 0.5 时各物体类别的 PR 曲线与非负坐标轴间围成的面积的均值），对于 mAP 值低的目标检测器，其物体检测的准确率和召回率都较低，因此通常不能正确地检测出物体的类别；而 mAP 值高的目标检测器不仅能准确地检测出物体的类别，还能尽可能多地将图像中可检测到的物体检测出来，因此使用 mAP 值这一指标能够很好地对目标检测器的整体表现进行评估。

此外，本工作认为，目标检测器能检测的物体种类数量对模型的导航性能可能也有一定的影响，因此本实验采取 mAP 值和检测物体种类数量作为目标检测器性能的衡量标准。为测试不同性能的目标检测器对导航表现的影响，实验准备了如表 4-6 所示的 7 个不同的目标检测器，其中模型 YOLOv5n、YOLOv5l、YOLOv5x 和 YOLOv5s 均为 YOLOv5 系列模型的一种，其中各种模型的复杂度、规模大小均不同。

表 4-6 目标检测器参数

目标检测器	模型	mAP	物体检测种类数量
检测器 1	YOLOv5s	0.124	16
检测器 2	YOLOv5s	0.53	80
检测器 3	YOLOv5s	0.62	16
检测器 4	YOLOv5s	0.17	16
检测器 5	YOLOv5n	0.46	80
检测器 6	YOLOv5l	0.67	80
检测器 7	YOLOv5x	0.69	80

其中所有这些检测器均在完整的 COCO2017 数据集[73]或 COCO2017 数据集的子集上进行预训练。其中对于物体检测种类数量为 16 的目标检测器，本工作在 COCO2017 数据集的 80 个物体检测种类种筛选出 16 个 AI2THOR 场景下出现频率可能较高的种类，并对数据集中包含这些样本的数据筛选出来，组成一个新的 COCO2017 子数据集，再由该子数据集训练得到目标检测器。检测器 2、4、5、6、7 则直接使用 YOLOv5 官方提供的预训练权重文件。

本实验在 AI2THOR 中的厨房和起居室两类场景中共选取 10 个仿真场景。对于每个实验场景，均随机选取 5 个目标点，对各个采用不同目标检测器的模型共进行 50 次实验。实验所用的模型预先经过了 2×10^7 步的训练。最终的实验结果采用平均轨迹长度和导航成功率这两个指标作为衡量标准，各模型的实验结果如表 4-7 所示（采用检测器 n 的模型称为模型 n）。

表 4-7 不同目标检测器对模型导航性能的影响

检测器	平均轨迹长度	导航成功率
检测器 1	807.3	52.2%
检测器 2	804.6	56.8%
检测器 3	715.3	56.6%
检测器 4	826.4	54%
检测器 5	790.7	56%
检测器 6	786.8	56%
检测器 7	733.3	57.2%

为准确找出目标检测器参数与模型导航表现间的关联性，如图 4-5 所示，绘制出 mAP 值、物体检测种类数量、平均轨迹长度和导航成功率间的关联图。

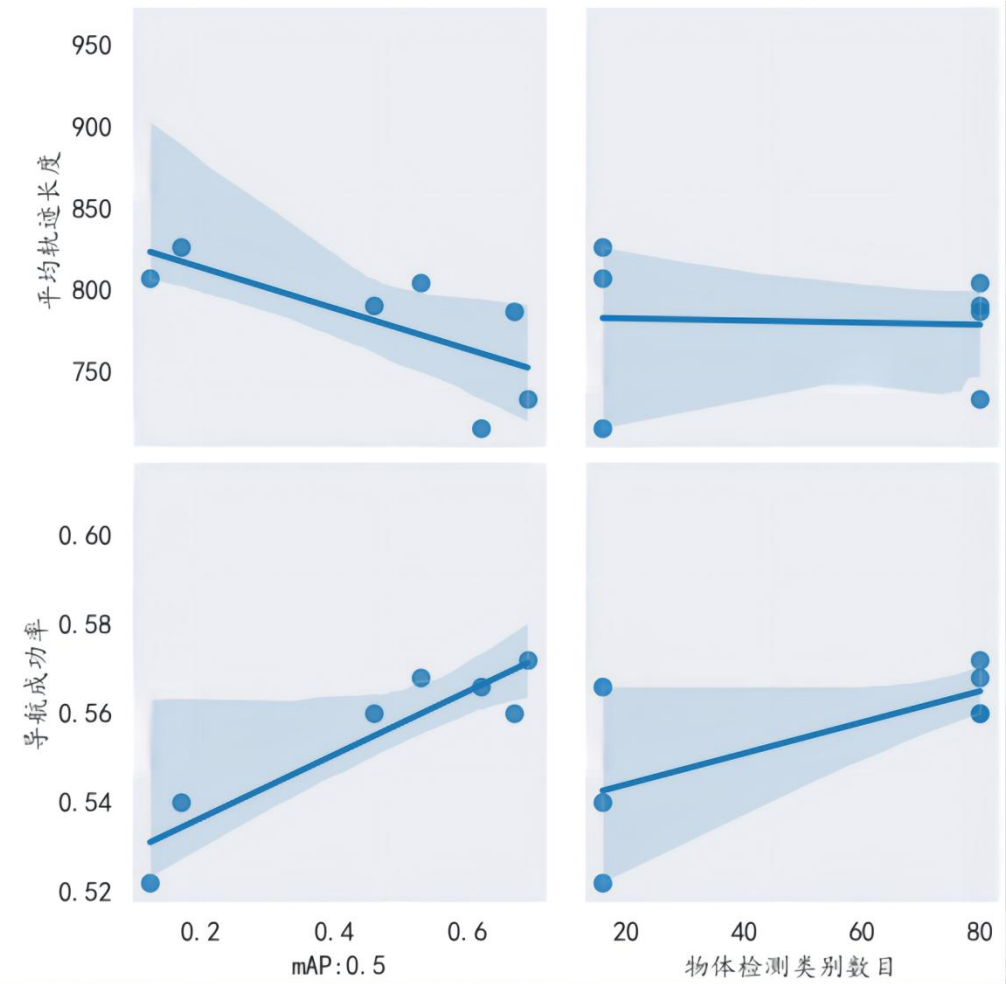

图 4-5 目标检测器性能对模型导航影响实验关联图

从关联图中能看出，mAP 值这一指标对模型导航表现有着较为显著的影响，使用高 mAP 值的目标检测器可以使模型的平均导航轨迹更短，导航成功率更高；而物体检测类别数目对模型的导航表现影响则较小，使用物体检测类别数目丰富的目标检测器可以使模型的导航成功率有轻微的提升，对模型的导航平均轨迹长度则无显著影响。

分析其中原因，本工作认为，高 mAP 值的目标检测器能够较为准确地识别出环境中的物体，而这些被准确检测出的物体可以为模型在导航过程中提供良好的定位功能，因此模型的整体导航表现得到了提高；而物体检测类别数目丰富的目标检测器有可能从目标点图像中识别出更多的目标物体，而数量丰富的目标物体可以为模型在导航过程中得到更多的定位参考，因此模型的导航成功率也随之得到提升。

综合以上实验结果，可以得出结论，使用高 mAP 值和物体检测种类丰富的检测器的模型能有相对更短的平均轨迹长度和相对更高的导航成功率。

4.2 讨论

4.2.1 本工作的创新点

针对传统的深度强化学习视觉目标导航工作存在跨场景泛化能力弱的问题，本文提出了一种新的深度强化学习视觉目标导航模型，实验表明该模型具有较好的跨目标泛化能力和跨场景泛化能力。相较于传统的深度强化学习视觉目标导航工作，本工作的主要创新点体现在以下三点：

1、提出了一种结合目标检测和深度图的状态表示方法，该表示方法能有效地减少训练时状态表示中包含的场景特有信息，对于模型跨场景泛化能力的提升有一定的帮助；

2、提出了一种结合目标检测结果的奖励函数表示方法；

3、将目标点信息以目标检测结果和奖励的形式融入到状态表示和奖励函数中，在避免在训练过程中对目标点图像进行重复处理的同时，也能保证模型的跨目标泛化能力。

4.2.2 目标检测器性能对模型的影响

除了 4.1.4 一节中实验所展示的影响以外，目标检测器对于模型导航表现的影响还包含很多其他的方面。在本节中对这些方面进行初步的探讨，探讨的内容也指引了对本工作改进的方向：

1、目标检测器无法从目标点图像中检测到目标物体的影响

由于本工作所提出的模型在导航过程中需要借助目标点图像中的目标物体进行定位，因此当目标检测器无法从目标点图像中检测到任何目标物体时，模型在导航过程中将无法有效进行定位，这将会对模型的导航表现产生极大的影响。

为验证这一猜想，本工作分别在 10 个仿真场景中选取了 10 个目标，并将目标分为数量相等的两组，在第一组目标中，目标检测器无法从目标检测模型中检测出任何目标物体（组 1）；在第二组目标中，目标检测器则能从目标检测模型中检测出一定数量的目标物体（组 2）。对每个目标均使用训练了 2×10^7 步的模型进行了 10 个回合的实验，为进行对比，实验使用随机游走模型对该 100 个目标分别进行了 10 个回合的实验。实验结果如表 4-8 所示。

表 4-8 实验结果

模型	平均轨迹长度	成功率
组 1	944.8	54.4%
组 2	733.9	68.8%
随机游走模型	1208.5	40.4%

从实验数据中能看出，组 1 的结果相较于组 2 的结果在平均轨迹长度和成功率这两个方面均相差了很多，这也基本上验证了这一猜想。

此外，从 4.1.4 一节的实验中，也看到了使用物体检测种类丰富的目标检测器的模型通常能得到一个较高的导航成功率。究其原因，本文认为是物体检测种类丰富的目标检测器从目标点图像中识别出目标物体的可能性更大，而这些丰富的目标检测结果信息为模型在导航过程中提供了良好的定位功能，模型也能更快地导航到目标点。

2、目标检测器仅能识别出物体的类别而并非物体实例

在采用判据一作为终止状态的判断依据的情况下，如果场景中出现多个属于相同类别的物体，并且模型从目标图像中检测出的物体数量较少时（例如只检测出一个物体时），模型有可能会出现目标混淆的现象，也即智能体最终到达的目标是场景中另一个同类物体的位置的现象。本文认为有以下几个可能原因：

（1）目标检测模块只能识别物体的种类而并非物体的实例；

（2）终止状态的判据不够严格，导致终止状态的误判；

（3）目标检测模块从目标点图像中识别的物体数量较少，模型没能学习到空间中物体间的位置关系。

3、目标检测器检测结果可能受遮挡、形变和光照等因素的影响

这些问题不仅仅是目标检测领域面临的问题，更是整个计算机视觉领域都在面临的问题。在实验中也观察到了某些物体在部分被遮挡后出现了无法被目标检测器所识别的情况，本文认为这种现象会导致模型需要更长的轨迹才能导航到目标点。如果能在这些问题上有所突破，能为模型的导航表现带来一定的提高。

4.3.3 模型导航表现稳定性分析

本文始终认为，好的目标导航模型应该能以较为稳定的表现完成导航任务。以传统的路径规划算法，如 Dijkstra 算法[76]、A*算法[77]等为例，每当导航起始点和目标点确定后，只要周围环境没有较大变化，导航路径会基本保持一致，这种稳定的导航表现才真正具有实际意义，目标导航模型也应该足够稳定地表现。

为分析本工作提出的模型的稳定性，实验在 AI2THOR 仿真环境中选取了一个规模中等的仿真场景。在该场景中，选取了一个固定的导航起始点，并且依次选取了三个与导航起始点的最短距离分别为 2 步、4 步和 8 步（以下分别称为近距离导航、中等距离导航和远距离导航）的目标点。在每次进行实验前，机器人都会被复位到固定的所选导航起始点。实验同时使用平均导航轨迹长度和导航成功率对模型的导航表现进行评估，由于所选场景较小，本实验中导航成功率的定义改为导航轨迹长度低于 300 步的轨迹占全部导航轨迹的比例。对每个目标点均进行 100 个回合的测试，由实验数据绘制得到的箱型图分别如图 4-6 和 4-7 所示。

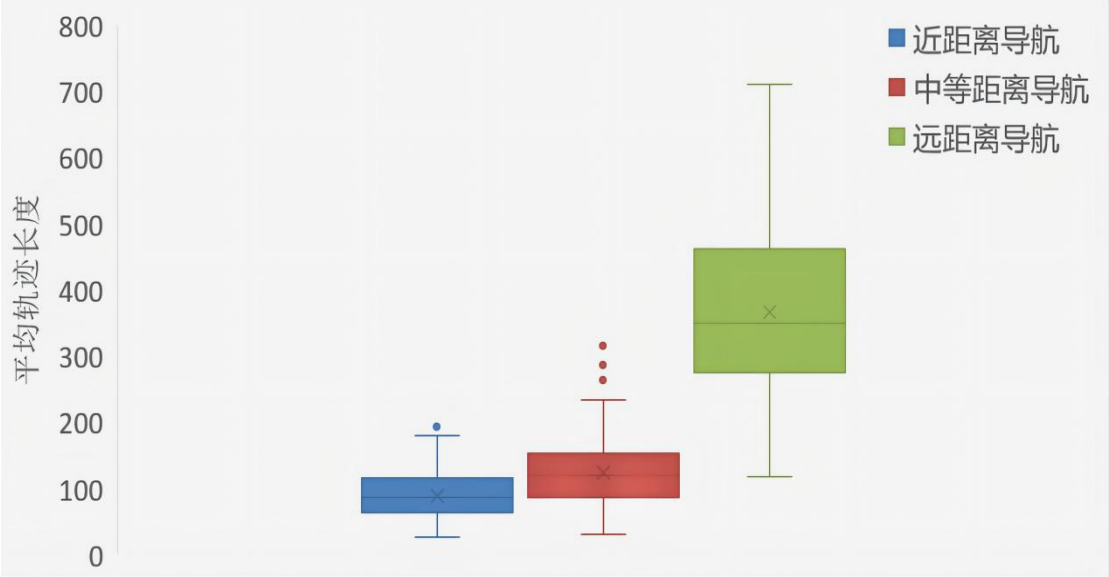

图 4-6 同一场景下的平均轨迹长度箱型图

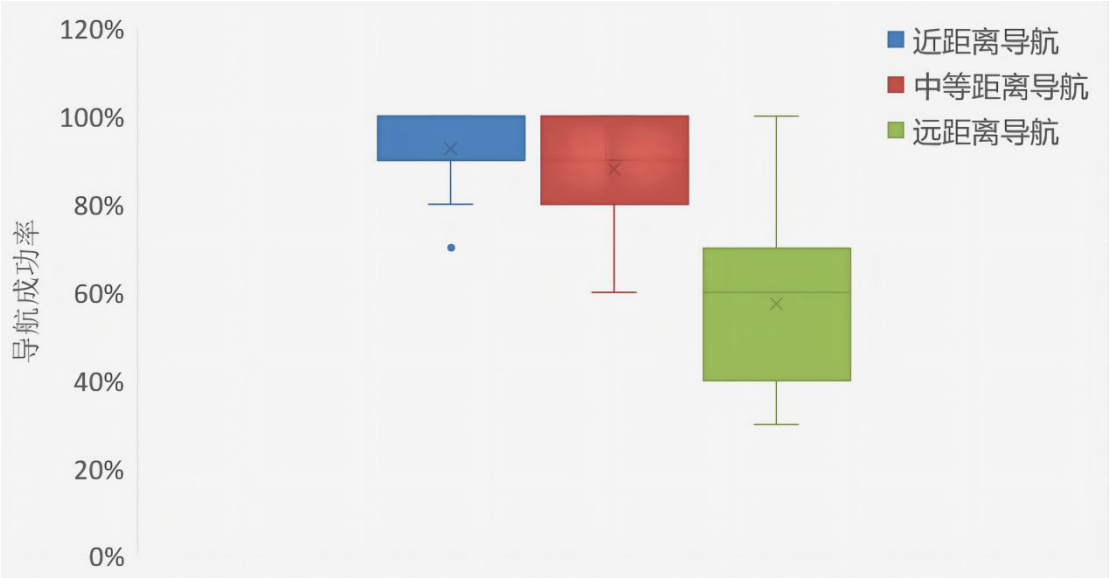

图 4-7 同一场景下的导航成功率箱型图

从实验结果能看到，无论是平均轨迹长度还是导航成功率，随着目标点与起始点距离的增加，其数据分布的方差会变得越来越大，这表明模型的导航表现醉着导航距离的变大而变得不够稳定。

此外，稳定的模型导航表现应当只与导航起始点和目标点间的距离、以及周围的环境分布有关，而不应该与所在环境的大小有关。实验在 AI2THOR 仿真环境上选取了小场景 FloorPlan408（状态空间大小为 44）、中等大小场景 FloorPlan4（状态空间大小为 80）和大场景 FloorPlan323（状态空间大小为 216）对模型进行测试。在每个场景当中均设置了一个固定的导航起始点，而导航的目标点与起始点间的距离固定为 8 步，对每个目标点，均进行了 100 个回合的测试。由实验数据绘制得到的箱型图如图 4-8 和图 4-9 所示。

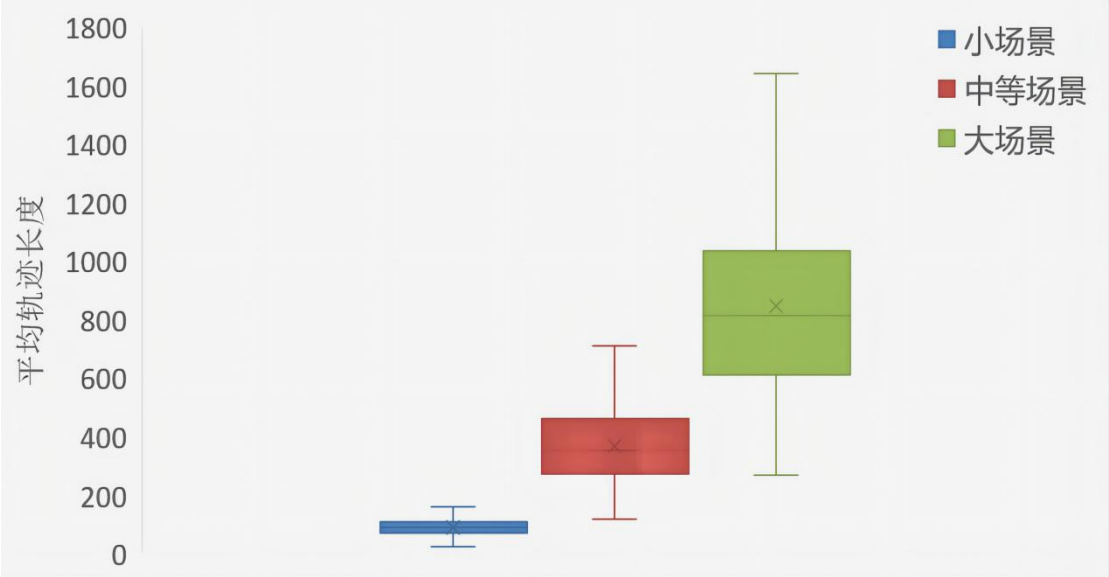

图 4-8 不同场景下的平均轨迹长度箱型图

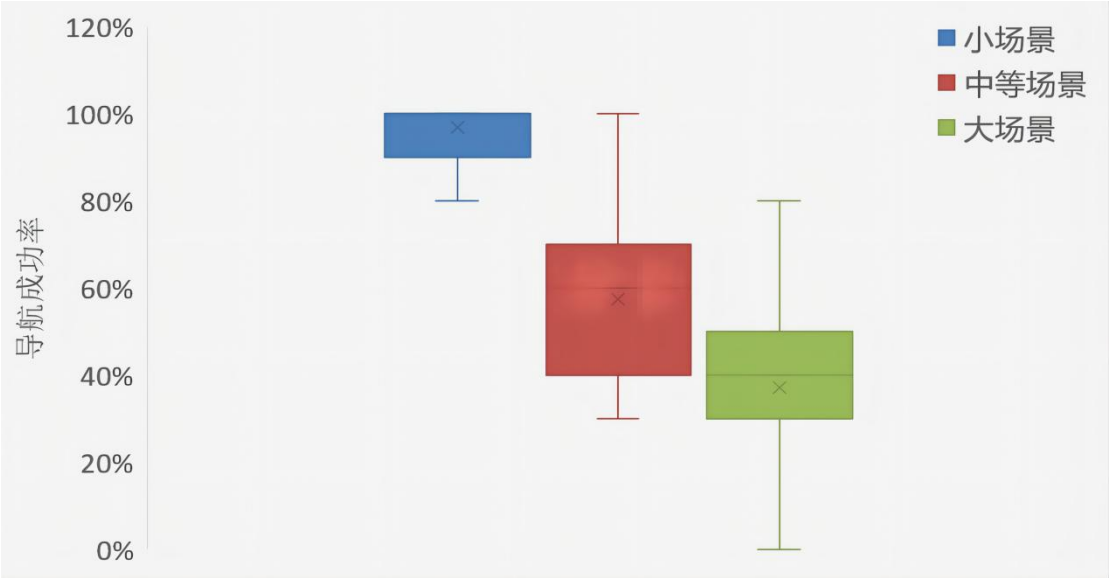

图 4-9 不同场景下的导航成功率箱型图

从实验结果中能看出，本模型的平均轨迹长度随着场景的增大而增加，而导航成功率则随着场景的增大而下降，并且随着场景增大，平均轨迹长度和导航成功率的数据分布方差也会变得越来越大，这表明模型的导航表现不够稳定。由此可见，本工作所提出的模型的导航表现还会受到场景大小这一因素的影响。

根据以上两个实验的结果，可以得出结论，目前本工作所提出的模型的导航表现收到了导航距离和导航成功率的影响，因此在导航表现稳定性这一方面上还存在着较大的改进空间。

4.4 本章小结

本章主要内容是离线 AI2THOR 数据集介绍以及对实验的设计和结果分析。在本章中，先对本文所提出的离线 AI2THOR 数据集进行了介绍，然后分别对模型的导航表现、跨目标泛化能力、跨场景泛化能力进行了实验并分析了实验结果，接下来分析了目标检测器性能对模型导航表现的影响，最后对本工作的创新点进行了总结，并针对模型的现有问题进行了较为深入的探讨，提供了一些本模型继续改进的思路。

结　论

随着人类社会的不断发展，人们对于机器人也提出了越来越高的要求，而机器人自主导航作为机器人完成很多任务的根本前提，在近些年来也逐渐受到了众多学者的关注。然而，传统的深度强化学习视觉导航方法，存在着跨场景泛化能力差的问题，本文的主要工作是针对该问题提出相应的解决方案。

本文对基于深度强化学习的机器人视觉目标导航方法进行了深入的探讨。在参考了 Y.Zhu 等人的工作[41]以及 A.Mousavian 等人的工作[43]中的思路后，本工作提出了一种新的基于深度强化学习的视觉目标导航模型。该模型将目标检测结果和深度图相结合构成当前时刻的状态表示，并将目标点图像的目标检测信息融合到状态表示和奖励函数中，在避免网络在训练过程中将单一目标点地信息隐式地融合到网络参数中的同时，也避免了对目标点图像的多次冗余处理。此外，为加快模型在 AI2THOR 仿真平台上的交互速度，本工作将 AI2THOR 中的所有室内仿真场景制作成了离线 AI2THOR 数据集，相较于直接使用 AI2THOR 仿真平台实时渲染，使用离线 AI2THOR 数据集能大大提高交互速度，并且能实现仿真环境的跨平台使用。

实验结果表明，本工作提出的模型具有较好的跨目标泛化能力和跨场景泛化能力，并且也有较高的数据效率；此外，目标检测器的性能对模型的导航表现具有一定的影响，因此一个好的目标检测器能有效地提高模型的导航表现；最后，本文探讨了模型导航表现的稳定性等问题，并认为目前本工作的模型并没有能达到对于好的导航表现的标准。

未来的工作包括但不限于以下几点：

（1）验证模型在真实环境中的表现。由于 AI2THOR 中的 RoboTHOR 仿真环境[29]具有较好的拟真效果，能模拟现实环境中的一些干扰，因此计划先将模型放到 AI2THOR 中的 RoboTHOR 环境中进行初步验证，之后再将模型部署到真实机器人中进行实验；

（2）验证目标检测奖励对模型的实际影响，本工作目前认为目标检测奖励的设置在一定程度上缓解了强化学习本身存在的稀疏奖励问题[78]，提高了本模型的学习效率和训练过程中的数据利用效率；

（3）解决在目标检测器无法从目标点图像中识别到任何目标物体时导致模型导航表现差的问题，本工作认为该问题是导致模型导航表现不稳定的原因之一，解决该问题有助于提高模型导航表现的稳定性。

参考文献

[1]梅荣娣.机器人在汽车制造领域中的应用分析[J].时代汽车,2022(07):33-34.

[2]2021 年中国商用服务机器人市场研究报告[J].机器人产业,2022(02):76-90.

[3]许雁容.医疗机器人改变医疗行业[J].机器人产业,2015(05):102.

[4]毛志贤,朱晓龙,韦建军,王春宝,刘铨权,段丽红,王同,罗承开,张广帅,王玉龙,龙建军,林焯华.家庭服务机器人现状与展望[J].机电工程技术,2021,50(02):8-14.

[5]李艳,鱼晨,戴庆瑜.室内主动式服务机器人控制系统研究与设计[J].陕西科技大学学报,2022,40(02):153-163.

[6]马凯,林义忠.移动机器人视觉导航技术综述[J].物流科技,2020,43(10):39-41+46.

[7]T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): part II," in IEEE Robotics & Automation Magazine, vol. 13, no. 3, pp. 108-117 (2006).

[8]Fuentes-Pacheco, J., Ruiz-Ascencio, J. & Rendón-Mancha, J.M. Visual simultaneous localization and mapping: a survey. Artif Intell Rev 43, 55–81 (2015).

[9]吴涛.用于移动机器人的视觉 SLAM 综述[J].数据通信,2022(01):48-51.

[10]R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction," in IEEE Transactions on Neural Networks, vol. 9, no. 5, pp. 1054-1054 (1998).

[11]闫皎洁,张锲石,胡希平.基于强化学习的路径规划技术综述[J].计算机工程,2021,47(10):16-25.

[12]杨思明,单征,丁煜,李刚伟.深度强化学习研究综述[J].计算机工程,2021,47(12):19-29.

[13]陈霖. 深度强化学习中的值函数研究[D].中国矿业大学,2021:1-84.

[14]多南讯,吕强,林辉灿,等. 迈进高维连续空间:深度强化学习在机器人领域中的应用[J]. 机器人,2019,41(2):276-288.

[15]段续庭,周宇康,田大新,郑坤贤,周建山.深度学习在自动驾驶领域应用综述[J].无人系统技术,2021,4(06):1-27.

[16]Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015):529-533.

[17]Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with Deep Reinforcement Learning[J]. arXiv preprint arXiv:1312.5602, 2013:1-9.

[18]Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." Nature 529.7587 (2016):484-489.

[19]Silver, David, et al. "Mastering the game of go without human knowledge." Nature 550.7676 (2017):354.

[20]Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J]. arXiv preprint arXiv:1509.02971, 2015:1-14.

[21]Haarnoja T, Tang H, Abbeel P, et al. Reinforcement learning with deep energy-based policies[C]//International Conference on Machine Learning. PMLR, 2017:1352-1361.

[22]Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]//International conference on machine learning. PMLR, 2018:1861-1870.

[23]Schulman J, Levine S, Abbeel P, et al. Trust region policy optimization[C]//International conference on machine learning. PMLR, 2015:1889-1897.

[24]Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J]. arXiv preprint arXiv:1707.06347, 2017:1-12.

[25]Ye D, Liu Z, Sun M, et al. Mastering complex control in moba games with deep reinforcement learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(04):6672-6679.

[26]Vinyals O, Ewalds T, Bartunov S, et al. Starcraft ii: A new challenge for reinforcement learning[J]. arXiv preprint arXiv:1708.04782, 2017:1-20.

[27]Kolve E, Mottaghi R, Han W, et al. Ai2-thor: An interactive 3d environment for visual ai[J]. arXiv preprint arXiv:1712.05474, 2017:1-4.

[28]Ehsani, Kiana et al. “ManipulaTHOR: A Framework for Visual Object Manipulation.” 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021):4495-4504.

[29]Deitke, Matt et al. “RoboTHOR: An Open Simulation-to-Real Embodied AI Platform.” 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020):3161-3171.

[30]Savva, Manolis et al. “Habitat: A Platform for Embodied AI Research.” 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019):9338-9346.

[31]Szot A, Clegg A, Undersander E, et al. Habitat 2.0: Training home assistants to rearrange their habitat[J]. Advances in Neural Information Processing Systems, 2021, 34: 1-16.

[32]Todorov E, Erez T, Tassa Y. Mujoco: A physics engine for model-based control[C]//2012 IEEE/RSJ international conference on intelligent robots and systems. IEEE, 2012: 5026-5033.

[33]James, Stephen et al. “RLBench: The Robot Learning Benchmark & Learning Environment.” IEEE Robotics and Automation Letters 5 (2020): 3019-3026.

[34]Gauci J, Conti E, Liang Y, et al. Horizon: Facebook's open source applied reinforcement learning platform[J]. arXiv preprint arXiv:1811.00260, 2018: 1-10.

[35]Levine S, Koltun V. Guided policy search[C]//International conference on machine learning. PMLR, 2013: 1-9.

[36]Kiran B R, Sobh I, Talpaert V, et al. Deep reinforcement learning for autonomous driving: A survey[J]. IEEE Transactions on Intelligent Transportation Systems, 2021: 1-18.

[37]Haotian Fu, Hongyao Tang, Jianye Hao, Zihan Lei, Yingfeng Chen, and Changjie Fan. 2019. Deep multi-agent reinforcement learning with discrete-continuous hybrid action spaces. In Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19). AAAI Press, 2329–2335.

[38]Y. Zheng et al., "Wuji: Automatic Online Combat Game Testing Using Evolutionary Deep Reinforcement Learning," 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2019, pp. 772-784.

[39]D.G.Treichler, Are you missing the boat in training aids?[J] Filem and Audio-Visual communication. 1967, 48(1): 14-16, 28-30, 48.

[40]Ye X, Yang Y. From seeing to moving: A survey on learning for visual indoor navigation (vin)[J]. arXiv preprint arXiv:2002.11310, 2020: 1-8.

[41]Y. Zhu et al., "Target-driven visual navigation in indoor scenes using deep reinforcement learning," 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 3357-3364.

[42]He, Kaiming et al. “Deep Residual Learning for Image Recognition.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016): 770-778.

[43]Mousavian, A., Toshev, A., Fiser, M., Kosecka, J., & Davidson, J. (2019). Visual

Representations for Semantic Target Driven Navigation. 2019 International Conference on Robotics and Automation (ICRA), 8846-8852.

[44]Lanctot M, Zambaldi V, Gruslys A, et al. A unified game-theoretic approach to multiagent reinforcement learning[J]. Advances in neural information processing systems, 2017: 1-14.

[45]Farebrother J, Machado M C, Bowling M. Generalization and regularization in DQN[J]. arXiv preprint arXiv:1810.00123, 2018: 1-29.

[46]Wortsman, M., Ehsani, K., Rastegari, M., Farhadi, A., & Mottaghi, R. (2019). Learning to Learn How to Learn: Self-Adaptive Visual Navigation Using Meta-Learning. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 6743-6752.

[47]Mirowski P, Pascanu R, Viola F, et al. Learning to navigate in complex environments[J]. arXiv preprint arXiv:1611.03673, 2016: 1-16.

[48]Andrychowicz M, Wolski F, Ray A, et al. Hindsight experience replay[J]. Advances in neural information processing systems, 2017: 1-11.

[49]Riedmiller M, Hafner R, Lampe T, et al. Learning by playing solving sparse reward tasks from scratch[C]//International conference on machine learning. PMLR, 2018: 4344-4353.

[50]Chen, Y., Liu, M., Everett, M., & How, J.P. (2017). Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. 2017 IEEE International Conference on Robotics and Automation (ICRA), 285-292.

[51]Chen, Y., Everett, M., Liu, M., & How, J.P. (2017). Socially aware motion planning with deep reinforcement learning. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1343-1350.

[52]Everett, M., Chen, Y., & How, J.P. (2018). Motion Planning Among Dynamic, Decision-Making Agents with Deep Reinforcement Learning. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 3052-3059.

[53]Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning[C]//International conference on machine learning. PMLR, 2016: 1928-1937.

[54]Geoffrey E. Hinton, Simon Osindero, Yee-Whye Teh; A Fast Learning Algorithm for Deep Belief Nets. Neural Comput 2006; 18 (7): 1527–1554.

[55]Minar M R, Naher J. Recent advances in deep learning: An overview[J]. arXiv preprint arXiv:1807.08169, 2018: 1-31.

[56]Rumelhart, D., Hinton, G. & Williams, R. Learning representations by back-propagating errors. Nature 323, 533–536 (1986).

[57]Babaeizadeh M, Frosio I, Tyree S, et al. Reinforcement learning through asynchronous advantage actor-critic on a gpu[J]. arXiv preprint arXiv:1611.06256, 2016: 1-12.

[58]Girshick, Ross B. et al. “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation.” 2014 IEEE Conference on Computer Vision and Pattern Recognition (2014): 580-587.

[59]Girshick R, Donahue J, Darrell T, Malik J. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. IEEE Trans Pattern Anal Mach Intell. 2016 Jan;38(1):142-58.

[60]Ren, Shaoqing et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (2015): 1137-1149.

[61]Redmon, Joseph et al. “You Only Look Once: Unified, Real-Time Object Detection.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016): 779-788.

[62]Redmon, Joseph and Ali Farhadi. “YOLO9000: Better, Faster, Stronger.” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017): 6517-6525.

[63]Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018: 1-6.

[64]Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020: 1-17.

[65]Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//European conference on computer vision. Springer, Cham, 2016: 21-37.

[66]Fu C Y, Liu W, Ranga A, et al. Dssd: Deconvolutional single shot detector[J]. arXiv preprint arXiv:1701.06659, 2017: 1-11.

[67]Liu S, Huang D. Receptive field block net for accurate and fast object detection[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 385-400.

[68]Zhang S, Wen L, Bian X, et al. Single-shot refinement neural network for object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition.

2018: 4203-4212.

[69]Ren, J., Chen, X., Liu, J., Sun, W., Pang, J., Yan, Q., Tai, Y.W., Xu, L.: Accurate single stage detector using recurrent rolling convolution. In: CVPR. (2017) 752-760

[70]Li Z, Zhou F. FSSD: feature fusion single shot multibox detector[J]. arXiv preprint arXiv:1712.00960, 2017: 1-10.

[71]Shen, Bokui et al. “iGibson 1.0: A Simulation Environment for Interactive Tasks in Large Realistic Scenes.” 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2021): 7520-7527.

[72]Paszke A, Gross S, Massa F, et al. Pytorch: An imperative style, high-performance deep learning library[J]. Advances in neural information processing systems, 2019, 32: 1-12.

[73]Lin T Y, Maire M, Belongie S, et al. Microsoft coco: Common objects in context[C]//European conference on computer vision. Springer, Cham, 2014: 740-755.

[74]Anderson P, Chang A, Chaplot D S, et al. On evaluation of embodied navigation agents[J]. arXiv preprint arXiv:1807.06757, 2018: 1-7.

[75]Batra D, Gokaslan A, Kembhavi A, et al. Objectnav revisited: On evaluation of embodied agents navigating to objects[J]. arXiv preprint arXiv:2006.13171, 2020: 1-9.

[76]E. W. Dijkstra. 1959. A note on two problems in connexion with graphs. Numer. Math. 1, 1 (December 1959), 269–271.

[77]P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths in graphs. IEEE Trans. Syst. Sci. and Cybernetics, SSC-4(2):100-107, 1968

[78]杨惟轶,白辰甲,蔡超,赵英男,刘鹏. 深度强化学习中稀疏奖励问题研究综述[J]. 计算机科学:1-13.