基于强化学习的机器人自主探索与物体感知算法

吴关, 夏熙, 曹合智, 刘利刚*

(中国科学技术大学数学科学学院 合肥 230026)
(lgliu@ustc.edu.cn)

摘 要: 针对如何在未知室内场景的探索中高效感知物体的问题, 提出一种机器人自主探索与物体感知算法. 利用深度强化学习让机器人通过与环境交互的方式学会利用场景的布局规律和语义信息获得更加高效、高质量的探索策略. 算法使用模块化的方式解决强化学习训练困难的问题, 分为同时定位与地图构建模块、全局探索模块、路径规划模块和局部探索模块. 首先同时定位与地图构建模块根据传感器所得数据构建地图; 然后全局探索模块根据当前地图决策长期目标点, 规划机器人将要探索的区域; 接着路径规划模块根据机器人当前位置和长期目标点规划行进路径; 最后局部探索模块基于机器人周围的局部地图信息规划每一步行进时传感器的朝向并更新地图. 在 Habitat 仿真环境中与 SC 和 ANS 这 2 种先进算法在 Gibson 和 Matterport3D 公开数据集上进行实验的结果表明, 所提算法在小、中、大和超大场景中的物体感知率分别为 0.942, 0.866, 0.652 和 0.506, 表现出对场景良好的感知性能.

关键词: 自主探索; 深度强化学习; 物体感知; 模块化框架
中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2024.20120

Autonomous Exploration and Object Perception Algorithm for a Robot Based on Reinforcement Learning

Wu Guan, Xia Xi, Cao Hezhi, and Liu Ligang*

(*School of Mathematical Sciences, University of Science and Technology of China, Hefei* 230026)

Abstract: In order to efficiently perceive objects in the exploration of unknown indoor scenes, an autonomous exploration and object perception algorithm for a robot is proposed. Using deep reinforcement learning, the robot learns to use the layout rules and semantic information of the scene to obtain a more efficient and high-quality exploration strategy through interaction with the environment. The algorithm uses a modular framework to conquer the difficulty in reinforcement learning training, which is divided into simultaneous localization and mapping module, global exploration module, path planning module and local exploration module. The simultaneous localization and mapping module construct a map based on the data obtained by sensors. Then the global exploration module decides a long term goal based on the map to guide the robot to the area to be explored. Next, the path planning module is employed to generate a collision-free trajectory for robot navigation. And the local exploration module plans the orientation of the sensor on robot at each step based on the local map and updates the map. The comparative experiments are conducted in Gibson and Matterport3D public datasets with SC and ANS two advanced algorithms in Habitat simulation environment. The results show that the proposed algorithm has object perception rates of 0.942, 0.866, 0.652 and 0.506 in small, medium, large and extra-large scenes respectively, demonstrating good perception performance for scenes.

收稿日期: 2023-01-05; 修回日期: 2023-04-14. **基金项目**: 国家自然科学基金(62025207); 国家重点研发计划(2022YFB3303400). **吴关**(1998—), 女, 硕士研究生, 主要研究方向为计算机图形学; **夏熙**(1993—), 男, 博士, 主要研究方向为计算机图形学; **曹合智**(1997—), 男, 博士研究生, 主要研究方向为计算机图形学; **刘利刚**(1975—), 男, 博士, 博士生导师, CCF 杰出会员, 论文通信作者, 主要研究方向为计算机图形学.

Key words: autonomous exploration; deep reinforcement learning; object perception; modular framework

随着机器人硬件的快速发展, 机器人开始广泛地出现于各种室内场景并给人们的生活带来智能服务. 与此同时, 其相关的算法研究引起了计算机图形学等诸多方向研究者的兴趣. 当机器人走入一个未知的室内场景时, 高效且高质量地感知到尽可能多的物体能让机器人更好地完成诸如导航到指定位置、导航到特定物体或场景等下游任务. 深度强化学习(deep reinforcement learning, DRL)[1-2]通过让机器人与环境交互的方式自主学习探索策略. 现有的基于 DRL 的自主探索算法大致分为端到端的算法和模块化的算法 2 种. 相较于端到端的学习算法, 模块化的算法能更好地利用场景的语义信息和布局规律学习到长期行为. 它将探索过程分解为全局模块、局部模块等不同模块, 每个模块侧重于解决不同的问题. 现有工作中, 全局模块致力于做出更高级的决策, 如决策下一个子目标点[3-4]、下一个边界点[5]、下一个图顶点[6-7]等. 局部模块致力于根据全局模块的高级决策选择机器人接下来的基础动作. 本文观察到, 人类在探索一个未知的室内场景时, 除了会全局地决策一个长期目标点, 也会在途中局部地调整眼睛的视角来获取更多的信息. 在前往长期目标点的过程中, 机器人周围也同样分布着许多物体, 但现有工作中传感器的朝向始终和机器人朝向保持一致, 均没有考虑在此过程中传感器朝向的选择, 导致现有的模块化算法在探索算法结束之后, 仍存在不少感知不全甚至没有被感知到的物体. 然而这些物体是不少下游任务中不可或缺的一环. 与此同时, 现有工作中机器人在探索时由于传感器朝向位置不当导致语义分割网络识别错误的现象也阻碍了部分需要精确语义信息的任务. 本文在局部模块中优化传感器的朝向, 让机器人在前往长期目标点的过程中通过选择合适的传感器朝向, 在获取更多物体语义信息的同时提高语义分割网络的识别准确率, 从而有望改善和加速对未知环境的语义感知.

1　相关工作

自主探索未知室内场景是机器人领域的一个重要问题, 早在 20 多年前就受到了研究人员的关注[8]. 传统算法使用启发式算法, 如将探索过程用部分可观测马尔可夫决策过程进行建模, 并使用优化算法近似求解[9-10]或基于边界的算法[11-12]让机器人不断走到最近的边界点, 从而持续扩展地图边界等. 然而启发式算法难以根据具有不确定性的未知环境提前制定最优策略, 并且无法利用可以减小搜索空间的语义先验.

相比于传统算法, 基于深度学习的算法能更好地建模室内场景的布局规律和语义信息, 并且对传感器噪声等误差更加鲁棒[13]; 不同于监督学习需要额外地构造监督数据[14], DRL 通过让机器人与环境交互的方式自主学习探索策略. 端到端的算法直接输入第 1 视角观测图和二维地图, 利用神经网络输出机器人下一步的基础动作, 如前进、左转、右转等. Tai 等[15]第 1 次在仿真器中搭建的小规模场景里利用 DQN(deep Q-networks)算法[16], 并使用原始传感器数据作为输入, 通过端到端的 DRL 来构建移动机器人的探索策略. Chaplot 等[17]利用端到端的算法使用物体语义信息和好奇心机制从机器人的探索策略中提炼一个有效的物体检测器. 由于机器人的探索行为需要长期连贯的行为, 如走出房间、穿过走廊. 但考虑强化学习奖励的稀疏性, 这种长期行为很难通过端到端的算法习得, 导致算法难以收敛. Chen 等[18]首先使用模仿学习进行预训练, 让机器人模仿人类的探索轨迹, 随后使用近端策略优化(proximal policy optimization, PPO)算法[19]进行强化学习训练. 然而即使具有基于模仿学习的预训练阶段, 端到端的策略仍需要用一个网络从数据中学习高效建图、状态估计和路径规划等多个目标, 其困难程度导致这类算法在一些方面依旧表现得不如完全不需要任何训练的传统算法. 为解决此问题, 模块化的算法将整个过程分解为不同模块. Chaplot 等[3]用模块化的方式将整体算法分为同时定位与地图构建(simultaneous localization and mapping, SLAM)模块、全局策略模块、路径规划模块和局部策略模块. SLAM 模块根据当前观测和上一步预测的地图来更新当前环境地图和估计机器人的姿态. 全局策略模块则以这两者作为输入规划长期目标点. 路径规划模块使用分析式的寻路算法进行路径规划并用启发式的算法将长期目标点转换为短期目标点. 局部策略模块根据单帧观测规划机器人下一步的基础动作以到达短期目标点. 在此工作基础上, Lodel 等[4]在局部策略模块中考虑了机器人动力学并加入了避免碰撞的硬约束, 最后采用优化的方

式进行求解. Chen 等[6]使用图来表征环境状态, 从而降低了状态空间的维度并有效提高了算法的泛化性. 本文在模块化算法的基础上, 提出了新颖的物体感知导向的自主场景探索算法.

2 本文算法概述

物体感知导向的自主探索流程如图 1 所示, 机器人通过一张实时更新的多通道全局俯视二维栅格地图 $M^{\mathrm{g}}_{t,i}$ 来维护它对场景自由区域、未知区域、障碍物区域以及物体语义信息的感知情况. 本文设定机器人初始位姿 $p_{0,0}$ 对应地图 $M^{\mathrm{g}}_{0,0}$ 的中心朝正北方, 并用其第 1 帧观测初始化全局地图. 在全局策略的第 t 步, 局部策略的第 i 步, 机器人的位姿估计为 $p_{t,i}$, 其获得来自传感器的深度图 $D_{t,i}$ 和 RGB 图片 $G_{t,i}$, SLAM 模块根据传感器的内参矩阵计算出 $D_{t,i}$ 在当前机器人局部坐标系下的点云, 并结合 Mask R-CNN(mask region-based convolutional neural network)语义分割网络[20], 从 $G_{t,i}$ 中获得的语义信息投影成自中心的语义二维地图 $M^{\mathrm{ego}}_{t,i}$; 随后利用当前位姿估计 $p_{t,i}$ 与初始位姿 $p_{0,0}$ 的相对位姿关系将 $M^{\mathrm{ego}}_{t,i}$ 整合进上一时刻的全局地图 $M^{\mathrm{g}}_{t-1,i}$, 得到更新后的全局地图 $M^{\mathrm{g}}_{t,i}$. 全局策略 $\Pi^{\mathrm{g}}_{\theta}$ 以 $M^{\mathrm{g}}_{t,i}$ 作为输入, 输出下一个长期目标点 a^{g}_{t+1} 指引机器人前往需要探索的区域. Chaplot 等[3]通过模仿学习获得局部运动策略让机器人前往长期目标点 a^{g}_{t+1}, 但他们在实验中发现, 基于学习的策略并没有表现出明显优于启发式策略的性能. 因而本文的路径规划模块直接使用了成熟的寻路算法 A*[21]来规划一条从上一个目标点 a^{g}_{t} 到新目标点 a^{g}_{t+1} 的无碰撞路径 R_t, 并采取一个启发式的运动策略将 R_t 拆分成一系列的由前进、向左、向右和停止 4 种机器人基础动作组成的动作集合 $\{a_{t,i}\}_i$, 机器人依次执行其中动作后能沿着规划路径无碰撞地移动到 a^{g}_{t+1}. 在机器人执行基础动作 $a_{t,i}$ 的同时, 局部策略 $\Pi^{\mathrm{l}}_{\theta}$ 截取全局地图 $M^{\mathrm{g}}_{t,i}$ 中以机器人为中心的一部分作为局部地图 $M^{\mathrm{l}}_{t,i}$, 并以此为输入; 而后输出此时机器人传感器转向动作, 并根据传感器最新观测更新全局地图 $M^{\mathrm{g}}_{t,i}$. 到达长期目标点 a^{g}_{t+1} 后, 全局策略继续规划新的目标点. 上述过程一直重复, 直到达到一个预先设定的步数阈值.

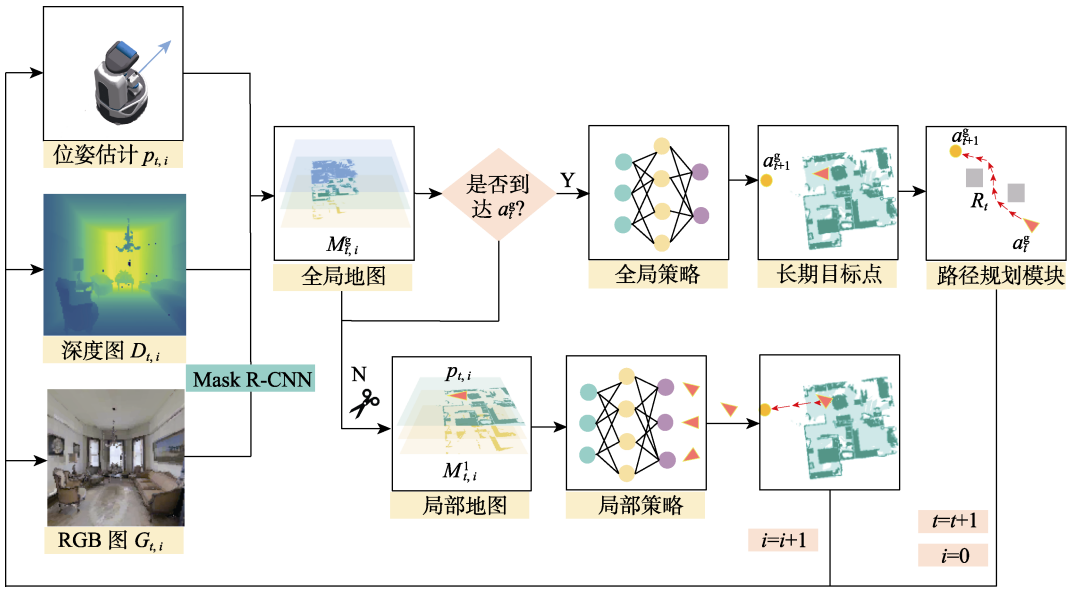


图 1 本文算法流程

本文算法步骤总结如下:

输入. 机器人初始位姿 $p_{0,0}$.

输出. 多通道全局地图 $M^{\mathrm{g}}_{t,i}$.

Repeat

SLAM 模块根据机器人当前观测的深度图 $D_{t,i}$, RGB 图 $G_{t,i}$ 和当前位姿 $p_{t,i}$ 更新全局地图 $M^{\mathrm{g}}_{t,i}$, $M^{\mathrm{g}}_{t,i} \leftarrow \mathrm{SLAM}\left(D_{t,i}, G_{t,i}, p_{t,i}\right)$.

全局策略 $\Pi^{\mathrm{g}}_{\theta}$ 根据当前全局地图 $M^{\mathrm{g}}_{t,i}$、机器人位姿 $p_{t,i}$ 和前一次输出的长期目标点 a^{g}_t 决策下一步的长期

目标点 a_{t+1}^{g}, $a_{t+1}^{\mathrm{g}} \leftarrow \Pi_{\theta}^{\mathrm{g}}\left(M_{t,i}^{\mathrm{g}}, p_{t,i}, a_t^{\mathrm{g}}\right)$.

路径规划模块根据机器人当前位姿 $p_{t,i}$ 和长期目标点 a_{t+1}^{g} 生成机器人从当前位置到达长期目标点的基础动作集合 $\left\{a_{t,i}\right\}_i$.

Repeat

以机器人为中心裁剪当前全局地图得到局部地图 $M_{t,i}^{\mathrm{l}}$.

局部策略 $\Pi_{\theta}^{\mathrm{l}}$ 根据当前局部地图 $M_{t,i}^{\mathrm{l}}$、深度图 $D_{t,i}$、机器人下一步动作 $a_{t,i+1}$ 和当前传感器朝向相对于机器人朝向夹角 β 的余弦值输出在机器人执行动作 $a_{t,i+1}$ 时传感器将会执行的转动动作 $a_{t,i+1}^{\mathrm{l}}$, $a_{t,i+1}^{\mathrm{l}} \leftarrow \Pi_{\theta}^{\mathrm{l}}\left(M_{t,i}^{\mathrm{l}}, D_{t,i}, a_{t,i+1} q, \cos\beta\right)$.

$i \leftarrow i+1$.

Until 机器人到达长期目标点 a_{t+1}^{g}.

$t \leftarrow t+1$, $i \leftarrow 0$.

Until 达到一个预先设定的步数阈值.

3　全局探索模块

全局探索模块根据室内场景布局规律和语义信息做出更加远期的决策. 其规划的长期目标点将指引机器人前往需要探索的区域. 全局探索模块的核心为全局策略. 接下来将介绍全局策略的观测空间和网络结构以及全局探索模块中用来优化该策略的奖励函数.

3.1　全局策略的观测空间

为了高效地探索未知室内场景, 机器人需要全面地掌握当前未知场景的探索进展. 为此, 本文将全局地图 $M_{t,i}^{\mathrm{g}}$ 表示成多通道的二维栅格地图, 共分成探索通道、障碍物通道、语义通道和轨迹通道 4 类.

3.1.1　探索通道

为了生成不重复的探索轨迹, 避免引导机器人多次前往相同的区域探索, 本文在全局地图中加入探索通道, 用来表示机器人已经探索过的区域. 具体来说, 在获得当前深度图并根据传感器内参矩阵计算出局部三维点云后, 在不移除地板平面的情况下, 直接将点云投影成一张值非 0 即 1 的栅格地图. 其中, 0 表示未探索过的区域; 1 表示探索过的区域.

3.1.2　障碍物通道

探索通道构建以后, 本文移除属于地板平面的点云并再次投影为栅格地图, 而后统计点云中的点落在各个格点里的数目 b. 随后利用非线性变换 $\mathrm{clip}\left(\left(b/b_{\max}\right)^2, 0, 1\right)$ 来归一化数值, 最终得到障碍物通道; 该数值可以近似作为该格点有障碍物的概率. 其中, $b_{\max}$ 是格点被认为完全不可穿越的阈值. 利用障碍物通道的障碍物分布, 机器人能学习到如何生成可到达的长期目标点, 并且能利用它来进行后续的路径规划.

3.1.3　语义通道

通过利用真实世界场景的布局规律并识别物体间的共存关系(如桌子椅子一般会同时出现), 机器人可以有效预见未扫描物体的存在. 结合移除属于地板平面的点云与 Mask R-CNN 识别 RGB 图片输出的语义信息, 最终得到带语义信息的点云. 本文筛选了 C_{s} 种室内常见的物件, 每一种类别物体的点云投影构成全局地图的一层语义通道, 每个像素记录该位置被投影到的次数, 即被 Mask R-CNN 识别为该类别的传感器帧数. 所有 C_{s} 种类别将构成 C_{s} 层语义通道. 其中, 一些不重要的物体种类如茶杯等全被归为“其他”类.

3.1.4　轨迹通道

为了让策略网络理解机器人是如何到达长期目标点的, 本文加入了对应机器人行进轨迹通道. 轨迹通道记录了机器人过去 n 步所走过的轨迹, 机器人当前所在位置为 1, 经过 n 步后线性衰减到 0. 通过线性衰减的方式, 该通道同时记录了机器人行进轨迹的时序信息.

最终全局地图 $M_{t,i}^{\mathrm{g}}$ 被表示为 $N^{\mathrm{g}} \times N^{\mathrm{g}} \times C^{\mathrm{g}}$ 的二维栅格地图. 其中, $N^{\mathrm{g}} \times N^{\mathrm{g}}$ 是全局地图分辨率, $C^{\mathrm{g}} = 3 + C_{\mathrm{s}}$, 即 $M_{t,i}^{\mathrm{g}} \in [0,1]^{N^{\mathrm{g}} \times N^{\mathrm{g}} \times C^{\mathrm{g}}}$. 为了更好地理解机器人自身位姿信息和上一轮目标点的空间关系, 全局策略还将以当前位姿 $p_{t,i}$ 和前一步的长期目标点 a_t^{g} 作为输入, 三者共同构成当前时刻全局策略的观测 $s_{t,i}^{\mathrm{g}} = \left\{M_{t,i}^{\mathrm{g}}, p_{t,i}, a_t^{\mathrm{g}}\right\}$.

3.2　全局策略的网络结构

图 2 展示了本文全局策略的网络结构. 本文使用去中心化分布式近端策略优化(decentralized distributed proximal policy optimization, DD-PPO)算法[22]作为 DRL 框架, 它是演员-评论员算法的一种. 演员-评论员算法是一种结合策略梯度和时序差分学习的强化学习算法. 其中, 演员是指策略函数 $\pi_{\theta}^{\mathrm{g}}\left(a_{t+1}^{\mathrm{g}} \mid s_{t,i}^{\mathrm{g}}\right)$, 它学习如何输出动作, 以在本回合中得到尽可能高的累计回报. 本文中将其表示为连续型随机策略, 输入当前环境状态 $s_{t,i}^{\mathrm{g}}$, 输出连续动作所属的高斯分布的均值 $\mu\left(s_{t,i}^{\mathrm{g}}\right)$ 和方差

$\sigma\left(s_{t,i}^{\mathrm{g}}\right)$，在此高斯分布 $N\left(\mu\left(s_{t,i}^{\mathrm{g}}\right),\sigma\left(s_{t,i}^{\mathrm{g}}\right)\right)$ 上采样得到最终输出的下一步长期目标点 a_{t+1}^{g}. 评论员是指状态价值函数 $V_{\pi}\left(s_{t,i}^{\mathrm{g}}\right)$，它对当前策略的未来累计回报进行预测，即评估演员的好坏. 借助价值函数，策略网络可以在任意指定步数后更新而不需要等到本回合结束才进行更新，从而加快了训练速度. DD-PPO 扩展了原始的 PPO 算法，其适用于本文这样在具有照片级真实感的三维场景等资源密集型的仿真环境中训练.

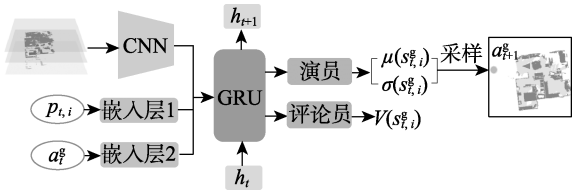


图 2 全局策略的网络结构

本文中演员和评论员共享同一个编码器来提取共同特征. 编码器的输入为 $s_{t,i}^{\mathrm{g}}$，其中的 $M_{t,i}^{\mathrm{g}}$ 经过卷积神经网络(convolutional neural network, CNN)，$p_{t,i}$ 和 a_t^{g} 分别经过 2 个嵌入层提取特征. 此外为了使全局策略知晓它所取得的进展避免重复的工作，本文引入了门控循环单元(gated recurrent unit, GRU)[23]. 其中的隐藏层状态 h_t 能为全局策略引入记忆力机制.

本文使用了 5 层的 CNN，每一层的结构参数如表 1 所示，每一层后使用 ReLU(rectified linear unit)作为激活函数，其输出展平后输入全连接层得到 512 维的特征向量. 嵌入层 1 和嵌入层 2 均为将相应输入映射为 32 维特征向量的全连接层. 这些特征向量经过拼接后输入隐藏层维度为 512 的单层 GRU 单元、输出 512 维的特征向量，再经过演员、评论员 2 个全连接网络，输出高斯分布的均值向量、方差向量和此时的价值函数值.

表 1 CNN 结构参数

层	卷积核大小、个数	步长
Cov1	8×8×13,32	4
Cov2	4×4×32,64	2
Cov3	4×4×64,64	2
Cov4	3×3×64,32	1
Cov5	3×3×32,16	1

3.3 奖励函数

本文将用来训练全局策略的奖励函数定义为

$$R_t = W_E^{\mathrm{g}}E_t + W_O^{\mathrm{g}}O_t + W_S^{\mathrm{g}}S_t + W_G^{\mathrm{g}}G_t.$$

其中，E_t 为探索面积项；O_t 为感知物体项；S_t 为懈怠惩罚项；G_t 为目标点可达项；W_E^{g}，W_O^{g}，W_S^{g}，W_G^{g} 分别为各自的组合系数.

探索面积项定义为机器人从 a_{t-1}^{g} 到 a_t^{g} 的过程中全局地图 $M_{t,i}^{\mathrm{g}}$ 探索通道中的增量

$$E_t = \sum_{(p,q)\in\left(N^{\mathrm{g}},N^{\mathrm{g}}\right)} l^2\left(E_{t+1,0}^{g}(p,q) - E_{t,0}^{g}(p,q)\right).$$

其中，$E_{t,0}^{\mathrm{g}}(p,q)$ 为 $M_{t,0}^{\mathrm{g}}$ 的探索通道 $E_{t,0}^{\mathrm{g}}$ 中 (p,q) 像素的值；l 为全局地图单个格点的实际边长.

感知物体项定义为机器人从 a_{t-1}^{g} 到 a_t^{g} 的过程中全局地图 $M_{t,i}^{g}$ 所有感知到物体的增量

$$O_t = \sum_{(p,q)\in\left(N^{\mathrm{g}},N^{\mathrm{g}}\right)} l^2 \operatorname{clip}\left(\sum_{k=1}^{C_{\mathrm{s}}} O_{t+1,0}^{\mathrm{g}}(k,p,q),0,f_{\max}\right)\Big/ f_{\max} - \sum_{(p,q)\in\left(N^{\mathrm{g}},N^{\mathrm{g}}\right)} l^2 \operatorname{clip}\left(\sum_{k=1}^{C_{\mathrm{s}}} O_{t,0}^{\mathrm{g}}(k,p,q),0,f_{\max}\right)\Big/ f_{\max}.$$

其中，$O_{t,0}^{\mathrm{g}}(k,p,q)$ 为全局地图 $M_{t,0}^{\mathrm{g}}$ 的所有语义通道中的第 k 层里 (p,q) 像素的值；$f_{\max}$ 为 (p,q) 像素被认为充分感知的阈值，即若传感器感知到 $f_{\max}$ 帧点云均投影至 (p,q) 像素，则认为该像素点被充分感知；l 为全局地图单个格点的实际边长.

懈怠惩罚项定义为机器人在执行基础运动动作 $a_{t,i}$(如前进、向左、向右)时所受到的惩罚. 即 S_t 为机器人从 a_{t-1}^{g} 到 a_t^{g} 的过程中执行的基础动作 $\{a_{t,i}\}_i$ 的总数目. 本文设定 $W_S^{\mathrm{g}}<0$，为了减少该惩罚，机器人需要避免不必要的来回走动.

目标可达项定义为全局策略决策出的长期目标点 a_t^{g} 由于指向墙的背面等因素造成该目标点没有路径可以到达所得到的惩罚 $G_t = l\left\|R_{t-1}^{-1} - a_t^{\mathrm{g}}\right\|$. 其中，$R_{t-1}^{-1}$ 表示机器人在规划路径 R_{t-1} 上到达的最后一个路径点；l 为全局地图单个格点的实际边长.

4 局部探索模块

路径规划模块是连接全局探索模块和局部探索模块的桥梁. 全局探索模块输出一个长期目标点 a_{t+1}^{g} 引导机器人前往需要探索的区域. 本文直接使用了成熟的寻路算法 A*[21]来规划一条从上一个目标点 a_t^{g} 到最新目标点 a_{t+1}^{g} 的无碰撞路径 R_t. 机器人根据当前在 R_t 上所在位置 $p_{t,i}$ 的切线方向 $d_{t,i}$ 和其自身朝向 $r_{t,i}$ 的夹角 $\alpha_{t,i}$ 与预先给定阈值 θ

的关系决定下一步的基础动作 $a_{t,i+1}$. 若 $\alpha_{t,i}<\theta$，则 $a_{t,i+1}$ 为前进；反之，$a_{t,i+1}$ 向左或向右转动使得 $\alpha_{t,i}$ 朝着小于 θ 的趋势变化. 运用此启发式运动策略得到的结果 $\{a_{t,i}\}_i$，能使得机器人从上一个目标点 a_t^{g} 沿着路径 R_t 无碰撞地行进到达新目标点 a_{t+1}^{g}，且前进时 $r_{t,i}$ 基本朝向 $d_{t,i}$ 方向.

局部探索模块在机器人执行基础运动动作 $a_{t,i}$ 的同时，决策机器人上传感器的转向动作 $a_{t,i}^{\mathrm{l}}$. 相比于行进时传感器一直和机器人朝向保持一致前进时朝着路径切线方向 $d_{t,i}$，优秀的传感器转向决策能让机器人在前往长期目标点的过程中感知到更多的物体，并使得语义分割网络更准确地识别出物体，从而加速机器人对未知环境的感知过程. 局部探索模块的核心为局部策略. 接下来将介绍局部策略的观测空间和网络结构，以及局部探索模块中用来优化该策略的奖励函数.

4.1　局部策略的观测空间

不同于全局策略需要综合考虑全局的探索进度和布局规律来决策下一个长期目标点，局部策略只需要根据机器人所在位姿 $p_{t,i}$ 周边的物体感知情况决策此刻传感器的朝向变化. 因此，尽管局部策略的观测空间有和全局策略的障碍物通道和 C_{s} 层语义通道一样构造的相应通道，但是为了防止观测空间里的无用成分在样本量有限时诱使 DRL 算法学习到虚假的决策相关性，并造成局部策略的过拟合，本文以机器人当前的位姿 $p_{t,i}$ 为中心，从分辨率为 $N^{\mathrm{g}}\times N^{\mathrm{g}}$ 的全局地图 $M_{t,i}^{\mathrm{g}}$ 中裁剪一块分辨率为 $N^{\mathrm{l}}\times N^{\mathrm{l}}$ 且具有 $C^{\mathrm{l}}=C_{\mathrm{s}}+1$ 层通道的自中心地图 $M_{t,i}^{\mathrm{l}}\left(\in[0,1]^{N^{\mathrm{l}}\times N^{\mathrm{l}}\times C^{\mathrm{l}}}\right)$，将其作为局部策略观测空间的一部分.

除自中心的局部地图 $M_{t,i}^{\mathrm{l}}$ 之外，场景中物体间的遮挡关系也应该在决策传感器朝向时被考虑进去，因此本文将深度传感器的观测数据 $D_{t,i}$ 一起作为输入. 同时，由于机器人自身的转动会导致传感器同步的转动，因此本文将机器人下一步的基础动作 $a_{t,i+1}$ 也作为输入. 假定机器人上的传感器相对于机器人每次转动的角度为 ω，最大转动幅度为 $[-\Omega,\Omega]$，其中的负号表示向左；则传感器最多连续向左或者向右转动 $[\Omega/\omega]$ 次，超过极限记为非法动作. 为了让局部策略学会减少输出非法动作，本文还将传感器相对于机器人转动角度 $\beta\left(\in[-\Omega,\Omega]\right)$ 的余弦值作为输入，则共同构成的当前局部策略的观测为 $s_{t,i}^{\mathrm{l}}=\left\{M_{t,i}^{\mathrm{l}},D_{t,i},a_{t,i+1},\cos\beta\right\}$.

4.2　局部策略的网络结构

与全局策略类似，局部策略同样使用 DD-PPO 算法[22]作为 DRL 算法框架. 局部策略的动作空间是离散型，分为传感器相对于机器人向左转、向右转和保持当前朝向静止不变 3 种动作. 因此它的策略函数为 $\pi_\theta^{\mathrm{l}}\left(a_{t,i+1}^{\mathrm{l}}\mid s_{t,i}^{\mathrm{l}}\right)$，即演员被设计为离散型随机策略，输出为选择每种离散动作的概率，根据此概率分布进行随机采样得到下一步传感器的转向动作 $a_{t,i+1}^{\mathrm{l}}$. 图 3 展示了局部策略的网络结构. 输入 $s_{t,i}^{\mathrm{l}}$ 中的局部地图 $M_{t,i}^{\mathrm{l}}$ 经过 CNN，深度图 $D_{t,i}$ 经过 ResNet-50[24]，$a_{t,i+1}$ 和 $\cos\beta$ 依次经过 2 个嵌入层提取特征，并最终通过 GRU 层进行整合. 提取到的通用特征经过演员输出动作，经过评论员输出未来累计回报的预测.

局部策略使用了和全局策略同样结构的 5 层 CNN. 深度图输入 ResNet-50 后展平接入全连接层输出 512 维特征向量. 嵌入层 1 和嵌入层 2 均为将

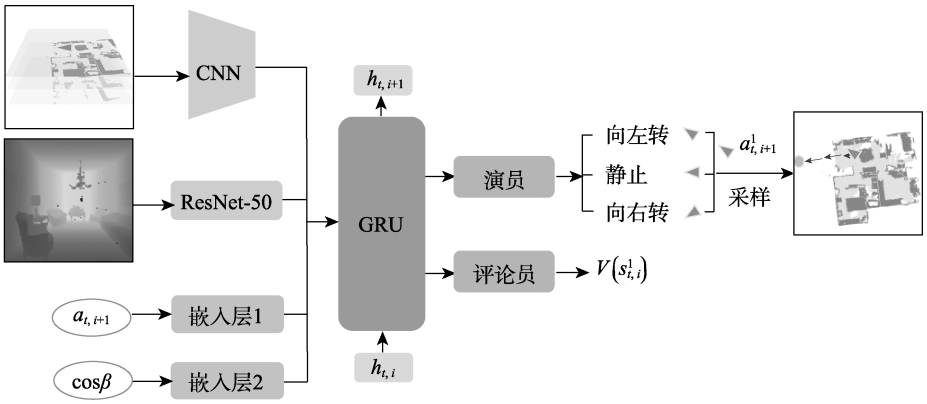


图 3　局部策略的网络结构

相应输入映射为 32 维特征向量的全连接层. 这些特征向量经过拼接后输入隐藏层维度为 512 的单层 GRU 单元、输出为 512 维的特征向量, 再经过演员、评论员 2 个全连接网络输出传感器动作的概率分布, 并预测此时价值函数值.

4.3 奖励函数

本文将用来训练局部策略的奖励函数定义为

$$R_{t,i}=W_O^1O_{t,i}+W_I^1I_{t,i}+W_S^1S_{t,i}.$$

其中, $O_{t,i}$ 为感知物体项; $I_{t,i}$ 为非法动作惩罚项; $S_{t,i}$ 为语义好奇心惩罚; W_O^1, W_I^1, W_S^1 分别为组合系数.

感知物体项与全局策略对应的奖励函数类似, 为一次局部探索动作中局部地图 $M_{t,i}^1$ 所有感知到物体的增量

$$O_{t,i}=\sum_{(p,q)\in(N^1,N^1)}l^2\,\mathrm{clip}\left(\sum_{k=1}^{C_\mathrm{s}}O_{t,i+1}^1(k,p,q),0,f_{\max}\right)\Big/f_{\max}-\sum_{(p,q)\in(N^1,N^1)}l^2\,\mathrm{clip}\left(\sum_{k=1}^{C_\mathrm{s}}O_{t,i}^1(k,p,q),0,f_{\max}\right)\Big/f_{\max}.$$

其中, $O_{t,i}^1(k,p,q)$ 为局部地图 $M_{t,i}^1$ 的所有语义通道中的第 k 层里 (p,q) 像素的值.

非法动作惩罚项定义了当局部策略输出非法动作时所给予的惩罚. 面对输出的非法动作, 传感器将不会执行该动作, 并保持相对于机器人的朝向静止不变. 输出非法动作时 $I_{t,i}$=1, 合法动作时 $I_{t,i}$=0. 本文设定 $W_I^1<0$, 所以为了减少惩罚, 局部策略需要根据输入的 β 角度信息避免输出非法动作.

语义好奇心惩罚定义了 Mask R-CNN 识别错误时给予的惩罚, 它鼓励传感器朝向合适的方向来使 Mask R-CNN 更加准确地识别出物体的类别. 本文将 $S_{t,i}$ 定义为所有语义通道每个像素位置信息熵增量的总和, 公式为

$$S_{t,i}=\sum_{(p,q)\in(N^l,N^1)}l^2\left(H_{t,i+1}^1(p,q)-H_{t,i}^1(p,q)\right).$$

其中, $H_{t,i}^1(p,q)=-\sum_{k=1}^{C_\mathrm{s}}p_{t,i}^1(k,p,q)\log\left(p_{t,i}^1(k,p,q)\right)$, $p_{t,i}^1(k,p,q)=\dfrac{O_{t,i}^1(k,p,q)}{\sum_{k=1}^{C_\mathrm{s}}O_{t,i}^1(k,p,q)}$, $O_{t,i}^1(k,p,q)$ 为局部地图 $M_{t,i}^1$ 的所有语义通道里的第 k 层中 (p,q) 像素里的值.

5 实 验

5.1 实验环境与参数设置

5.1.1 实验环境与数据集

本文使用 Habitat 平台[25]作为算法训练和测试的仿真环境, 它分为 Habitat-Sim 和 Habitat-API 这 2 个部分. 前者是一个高度可配置、灵活且高性能的三维仿真器, 通过设置机器人和传感器并利用通用的三维场景数据集来快速生成高真实感的 RGB 图片和深度图片. 后者是在 Habitat-Sim 的基础之上封装而成的模块化高级库, 定义了用于具身 AI 开发的不同任务(如导航任务、指令跟踪任务等)、与任务相关的传感器和机器人以及衡量任务进展的度量等. 由于本文算法的新颖性, Habitat-API 中没有现有的任务和相应的配置. 因此, 本文在 Habitat 环境中定义了全局探索任务和局部探索任务 2 个新任务, 并在此基础上进行训练和评估.

本文实验使用的室内场景数据来自于 Matterport3D(简称为 MP3D)数据库[26]和 Gibson 数据库[27], 这 2 个数据库中的场景均源于人工扫描的真实世界中的室内场景. 最终, 本文汇集了 111 个场景(MP3D 90 个, Gibson 21 个), 并用其中的 78 个场景(MP3D 61 个, Gibson 17 个)组成训练集, 33 个场景(MP3D 29 个, Gibson 4 个)组成测试集. 经过统计, 所收集的场景平均面积有 275.2 m^2, 并包含 22.3 个区域. 为了在评估时充分反映算法性能, 本文在测试时将依据可通行面积将测试集划分为 4 个子集: 小场景(9 个, 面积 17~97 m^2, 平均包含物体数 60.7 件)、中场景(10 个, 面积 100~224 m^2, 平均包含物体数 526.1 件)、大场景(8 个, 面积 264~391 m^2, 平均包含物体数 560 件)以及超大场景(6 个, 面积 405~1 371 m^2, 平均包含物体数 799.7 件).

5.1.2 实现细节

实验中, RGB 图片和深度图片分辨率均设置为128×128. 仿真环境假设 RGB 传感器和深度传感器的视点是一致的. 由于在真实场景中可以输出 RGB 图片流和深度图片流的硬件 Microsoft Kinect 传感器官方发布的开发人员工具包中提供了2种传感器视点相互转换的接口, 因而本实验在仿真环境中假设 2 种传感器视点一致具有一定的合理性.

全局策略输入中全局地图的分辨率为 N^g =

720, 每个格点的边长 l 为 5 cm, 因而能覆盖 36 m×36 m 的平面区域, 这对室内场景而言是较为充足的大小范围. 构造障碍物通道时使用的视作完全不可穿越的阈值为 $b_{\max}=10$. 语义通道中共筛选了 $C_{\mathrm{s}}=10$ 种室内常见物件, 全局地图的总通道数 $C^{\mathrm{g}}=13$. 轨迹通道记录了机器人过去 $n=100$ 步所走过的轨迹. 奖励函数中的组合系数是 DRL 中需要调节的超参数, 组合系数的选取需要让每一项能保持在一个相当的数量级, 在此基础上本文选取了使得下文中介绍的评判指标表现最好的一组参数, 分别为 $W_E^{\mathrm{g}}=0.05, W_O^{\mathrm{g}}=0.02, W_S^{\mathrm{g}}=-0.01, W_G^{\mathrm{g}}=-0.01$. 奖励函数感知物体项的计算中, 被认为充分感知的帧数阈值为 $f_{\max}=4$.

局部策略输入的局部地图是从全局地图中以机器人所在位置为中心, 所裁剪的大小为 12 m×12 m, 分辨率 $N^{\mathrm{l}}=240$ 的二维栅格地图. 机器人上传感器相对于机器人每次转动的角度设置为 $\omega=\dfrac{\pi}{9}$, 最大转动幅度为 $\Omega=\dfrac{4\pi}{9}$. 局部策略奖励函数的组合系数也同样是一组超参数, 其选取依据和全局策略类似, 本文分别设置为 $W_O^{\mathrm{l}}=0.04$, $W_I^{\mathrm{l}}=-0.1$, $W_S^{\mathrm{l}}=-0.05$. 本文使用点导航任务[22]训练得到的 ResNet-50 的权重来为局部策略相应模块进行初始化.

5.1.3　训练与测试

全局策略在一台配置了一块 Intel Xeon Platinum 8163 型号, 768 GB 的 48 核 CPU 内存和 8 块 NVIDIA Tesla V100 型号, 32 GB 显存的云服务器上, 训练时间大约为 1 d.

局部策略在一台配置了一块 AMD EPYC 7402 型号, 512 GB 内存的 48 核 CPU 和 2 块 NVIDIA GeForce RTX 3090 型号, 24 GB 显存的云服务器上, 训练时间大约为 2 d.

训练时, 本文先固定传感器相对于机器人朝向保持不变, 即采用前进时传感器朝向路径切线方向的方式训练全局策略; 再使用初步训练好的全局策略的权重训练局部策略; 最后将初步训练好的两者结合在一起进行策略网络的微调. 训练时都采用 Habitat PointNav 挑战赛[25]的回合数据集. 其中的每一个回合的数据指定了场景名称、机器人的初始位置和初始朝向信息, 每个场景中都有数十个回合. 通过把机器人放在同一个场景不同初始位置, 可以提升训练的样本量. 测试时, 将训练好的策略网络在一台配备有一块 AMD Ryzen 95 950X 型号, 64 GB 内存的 16 核 CPU 以及一块 24 GB 显存的 NVIDIA GeForce RTX 3090 GPU 的台式机上.

全局策略训练回报曲线如图 4 所示, 由于全局策略是连续型策略, 因此其训练难度更大, 收敛得也更加缓慢. 局部策略训练回报曲线如图 5 所示, 训练初期在非法动作惩罚和语义好奇心惩罚的作用下每回合平均回报值呈负值, 随着训练的进行回报值逐渐转为正值并趋于平稳.

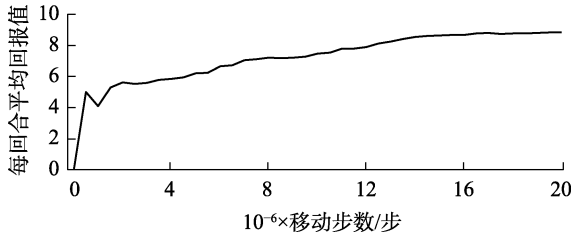


图 4　全局策略训练回报曲线图

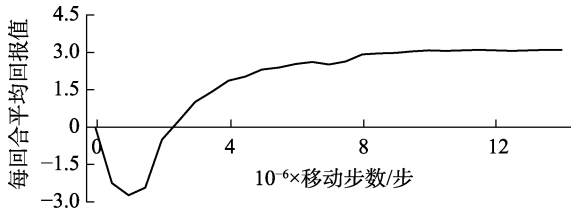


图 5　局部策略训练回报曲线图

5.2　性能比较

本文实验围绕 3 个问题展开: (1) 本文算法能否提高机器人感知物体的完整度? (2) 本文算法能否提高机器人感知物体的效率? (3) 本文算法能否提高机器人识别物体的准确率?

5.2.1　评价指标与基准算法

为了评估本文自主场景探索算法的性能, 本文使用以下评价指标:

(1) 物体感知率(object perception rate, OPR). 尽可能完整地感知室内物体和物体间的相对位置关系是自主场景探索算法需要实现的首要目标, 对导航到特定物体或位置等下游任务有重要意义. OPR 反映了被传感器感知到的物体占总物体的比例. 指定回合上限步数 T 后, 本文通过结束时全局地图所有语义通道 O_T 中每个对应像素被感知的概率之和与真实地图中所有语义通道 O^* 内所有像素值之和的比值来计算 OPR, 公式为

OPR =

$$\frac{\sum_{(p,q)\in(N^{\mathrm{g}},N^{\mathrm{g}})}\left(\mathrm{clip}\left(\sum_{k=1}^{C_s}O_T(k,p,q),0,f_{\max}\right)\Big/f_{\max}\right)}{\sum_{(k,p,q)\in(C_s,N^{\mathrm{g}},N^{\mathrm{g}})}O^*(k,p,q)}.$$

其中，$O^*(k,p,q)$ 表示根据数据集真实标签得到的所有语义通道中的第 k 层里 (p,q) 像素的值，第 k 类物体投影所占像素的值记为 1，其余值记为 0.

(2) 场景覆盖率(scene coverage rate, SCR). 它反映了已探索的场景占全部可探索场景的比例. 指定回合上限步数 T 后，通过结束时全局地图探索通道 E_T 中所有像素的值之和与真实地图探索通道 E^* 所有像素值之和的比值来计算 SCR，公式为

$$\mathrm{SCR}=\sum_{(p,q)\in(N^{\mathrm{g}},N^{\mathrm{g}})}E_T(p,q)\Big/\sum_{(p,q)\in(N^{\mathrm{g}},N^{\mathrm{g}})}E^*(p,q).$$

(3) 移动步数. 高效的自主探索算法应该避免不必要的来回走动，尽可能快速地完成对场景的感知. 本文通过统计完成固定的 OPR 的移动步数来衡量自主探索算法的效率.

(4) 物体识别准确率(object detection accuracy, ODA). 提高 Mask R-CNN 的 ODA 对于部分需要精确语义信息的任务有重要意义. 它反映了类别识别正确的物体占所有被传感器感知物体的比例. 指定回合上限步数 T 后，本文通过结束时全局地图的所有语义通道 O_T 中，Mask R-CNN 预测正确的概率值之和与每个对应像素被感知的概率之和的比值来度量来计算 ODA，公式

ODA =

$$\frac{\sum_{(k,p,q)\in(C_s,N^{\mathrm{g}},N^{\mathrm{g}})}\min\left\{\frac{O_T(k,p,q)}{\sum_{k=1}^{C_s}O_T(k,p,q)},O^*(k,p,q)\right\}}{\sum_{(p,q)\in(N^{\mathrm{g}},N^{\mathrm{g}})}\left(\frac{\mathrm{clip}\left(\sum_{k=1,}^{C_s}O_T(k,p,q),0,f_{\max}\right)}{f_{\max}}\right)}.$$

将本文算法与基于 DRL 的端到端的 Semantic Curiosity(简称为 SC)[17]算法和模块化的 Active Neural SLAM(简称为 ANS)[3]算法进行定量比较. 实验中，所有算法都训练了 2 000 万步.

5.2.2 物体感知完整度

表 2 中列举了 3 种算法在不同场景大小的测试集中的性能表现. 实验中的 OPR 和 SCR 为指定步数上限时(小、中场景 1 000 步，大、超大场景 1 500 步)获得的平均结果.

本文策略以最大化 OPR 为目标，从表 2 中可以看出，本文算法在不同大小的场景中都表现出了最优的 OPR，验证了其对提高机器人感知物体完整度的有效性. ANS 以最大化 SCR 为目标，确实取得了最优的表现，但它探测了不少没有物体的区域，导致 OPR 较低. SC 算法在优化目标中引入了语义好奇心机制，鼓励机器人从多个角度观察物体，最终在小、中场景中表现出了优于 ANS 算法的 OPR. 但由于端到端的 SC 算法缺少长期目标点的指引，难以习得长期连续的行为，因此在大、超大场景中的 SCR 表现欠佳. 这种探索的不足导致 SC 算法整体的 OPR 下降，最终其在大、超大场景中的 OPR 略低于 ANS 算法. 本文算法通过全局策略和局部策略，兼顾长期目标点指引探索和合适的传感器朝向提高物体感知能力，具有能更完整地感知未知场景里物体的性能.

表 2 3 种算法在指定步数上限下的对比

算法	小场景		中场景		大场景		超大场景	
	OPR	SCR	OPR	SCR	OPR	SCR	OPR	SCR
SC	0.877	0.806	0.807	0.724	0.566	0.487	0.416	0.343
ANS	0.859	0.956	0.792	0.885	0.584	0.671	0.448	0.520
本文	0.942	0.883	0.866	0.816	0.652	0.627	0.506	0.473

5.2.3 物体感知加速

本文使用移动步数来衡量物体感知的效率. 移动步数为达到指定 OPR(小场景 0.8、中场景 0.7，大场景 0.5，超大场景 0.4)时的平均结果.

图 6 中可以看出在达到相同 OPR 时，本文算法的移动步数明显更少. 这表明本文算法的物体感知效率优于 2 种对比算法，能更快速地感知未知场景中的物体. ANS 算法会耗费时间去探索物体稀疏的区域. 端到端的 SC 策略直接输出下一个基础动作，如左转、右转和前进，过细的决策粒度导致其难以连续生成相当数量的无碰撞动作来移动到需要探索的区域，因而也表现出低效性，并且这种低效性在大和超大场景中体现得更为显著.

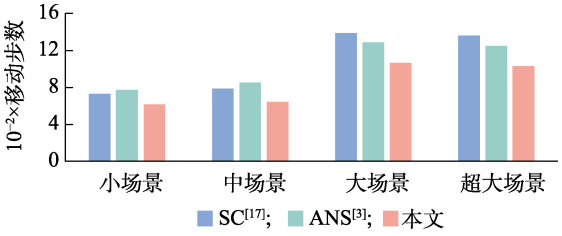


图 6 不同算法达到指定 OPR 的移动步数对比

5.2.4　ODA

图 7 中展示了 3 种算法在不同场景大小的测试集中 Mask R-CNN 的 ODA 性能表现. 实验中的 ODA 为到达指定步数上限时(小、中场景 1 000 步，大、超大场景 1 500 步)获得的平均结果. 从图 7 中可以看出，本文算法能显著地提高 Mask R-CNN 的 ODA. SC 策略的语义好奇心机制鼓励其探索能够增加语义信息的区域，在探索中传感器更倾向于从不同的视角对准物体，因而 Mask R-CNN 也能较准确地识别出一些物体；但这些角度并不总是能让传感器正视物体，因而在指定步数内其 ODA 表现得不如本文算法. ANS 算法以鼓励覆盖场景最大化为导向，其传感器倾向于探索未知区域，导致其并不总是朝着更利于识别出物体的方向，在 3 种算法的探索轨迹中 ODA 最低.

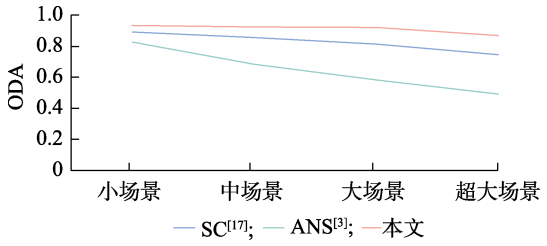


图 7　不同算法达到指定移动步数的 ODA 对比

5.3　消融实验

本文设计的局部探索模块在机器人前往长期目标点的过程中，在不改变机器人朝向的前提下规划传感器的转向动作，使传感器转向能感知到更多物体并能更准确地识别出物体的方向，提升机器人对整体场景的语义理解. 为了验证其必要性，本文去除局部探索模块，即仅使用全局探索模块决策下一个将要探索的长期目标点，并用寻路算法 A*规划行进路线，传感器相对于机器人固定不动，因而总是朝向路径切线方向前进.

5.3.1　局部探索模块对 OPR 的影响分析

本文首先在实验中对比去除和不去除局部探索模块 2 种算法在不同场景大小的测试集中探索指定步数上限(小、中场景 1 000 步，大、超大场景 1 500 步)时获得的 OPR 的平均结果.

对比图 8 中的结果可以看到，局部探索模块在不同大小规模的未知场景中都对提高 OPR 有显著的帮助. 局部探索模块能让机器人通过转动传感器朝向来感知到更多的物体；同时这也将产生信息量更充足的地图，有利于配合全局探索模块决策指向场景里未被充分感知区域的长期目标点. 去除局部探索模块以后，传感器朝向总是固定和

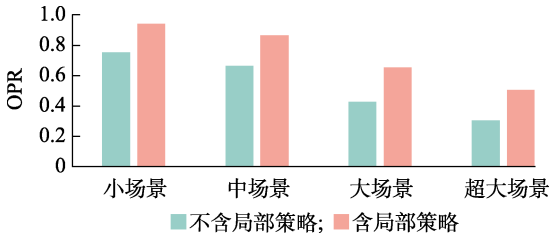


图 8　局部探索模块对 OPR 影响消融对比

路径切向保持一致，这种不灵活的方式使得机器人难以全面感知路途中经过的物体，最终表现出较低的 OPR.

5.3.2　局部探索模块对 ODA 的影响分析

去除和不去除局部探索模块 2 种算法在不同场景大小的测试集中探索指定步数上限(小、中场景 1 000 步，大、超大场景 1 500 步)时 Mask R-CNN 的 ODA 的平均结果，如图 9 所示.

对比图 9 中的结果可以看到，局部探索模块对提高 Mask R-CNN 的 ODA 有着较显著的作用. 局部探索模块能让机器人在前往全局探索模块规划的长期目标点的过程中，通过转动传感器朝向到更有利于识别正确的角度，如正视物体的方向，来提高 Mask R-CNN 的 ODA.

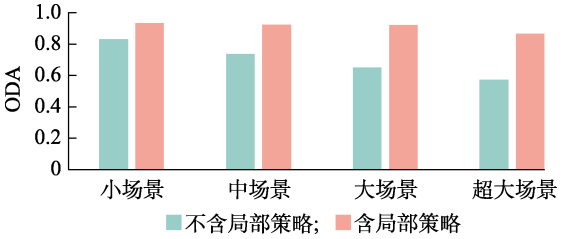


图 9　局部探索模块对 ODA 影响消融对比

6　结　语

本文提出了一种物体感知导向的机器人自主探索算法，该算法是基于 DRL 的模块化自主探索算法，分为 SLAM 模块、全局探索模块、路径规划模块和局部探索模块. SLAM 模块用来生成全局和局部策略观测空间中的全局和局部地图；全局探索模块根据全局地图决策长期目标点，该长期目标点决定了机器人下一步去探索的区域；路径规划模块生成从当前位置到长期目标点的无碰撞路径，并确定能让机器人沿着规划路径行进的一系列基础动作；局部探索模块决策机器人行进过程中传感器相对于机器人的转向动作. 相比于其他先进的替代算法，本文算法能显著地提高 OPR、感

知效率和语义分割网络的 ODA.

本文仅在仿真平台中验证了所提算法. 从仿真实验到真实场景的推广是具有潜力并值得未来进一步研究的方向; 同时利用多智能体和强化学习的组合设计分散的协作算法, 是机器人自主探索中的另一个具有潜力并充满挑战的方向.

参考文献(References):

[1] Arulkumaran K, Deisenroth M P, Brundage M, *et al*. Deep reinforcement learning: a brief survey[J]. IEEE Signal Processing Magazine, 2017, 34(6): 26-38

[2] Yang Siming, Shan Zheng, Ding Yu, *et al*. Survey of research on deep reinforcement learning[J]. Computer Engineering, 2021, 47(12): 19-29(in Chinese)
(杨思明, 单征, 丁煜, 等. 深度强化学习研究综述[J]. 计算机工程, 2021, 47(12): 19-29)

[3] Chaplot D S, Gandhi D, Gupta S, *et al*. Learning to explore using active neural SLAM[OL]. [2023-01-05]. https://arxiv.org/abs/2004.05155

[4] Lodel M, Brito B, Serra-Gómez Á, *et al*. Where to look next: learning viewpoint recommendations for informative trajectory planning[C] //Proceedings of the IEEE International Conference on Robotics and Automation. Los Alamitos: IEEE Computer Society Press, 2022: 4466-4472

[5] Niroui F, Zhang K C, Kashino Z, *et al*. Deep reinforcement learning robot for search and rescue applications: exploration in unknown cluttered environments[J]. IEEE Robotics and Automation Letters, 2019, 4(2): 610-617

[6] Chen F F, Martin J D, Huang Y W, *et al*. Autonomous exploration under uncertainty via deep reinforcement learning on graphs[C] //Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Los Alamitos: IEEE Computer Society Press, 2020: 6140-6147

[7] Lu Yazhou, Huang Jing, Ruan Xiaogang, *et al*. A robot autonomous exploration method combining graph neural network and deep reinforcement learning[J/OL]. Control and Decision: 1-9[2023-01-05]. https://doi.org/10.13195/j.kzyjc.2021.1909(in Chinese)
(鲁亚洲, 黄静, 阮晓钢, 等. 结合图神经网络和深度强化学习的机器人自主探索方法[J/OL]. 控制与决策: 1-9 [2023-01-05]. https://doi.org/10.13195/j.kzyjc.2021.1909)

[8] LaValle S M. Rapidly-exploring random trees: a new tool for path planning[OL]. [2023-01-05]. https://www.semanticscholar.org/paper/Rapidly-exploring-random-trees-%3A-a-new-tool-for-LaValle/d967d9550f831a8b3f5cb00f8835a4c866da60ad

[9] Martinez-Cantin R, de Freitas N, Brochu E, *et al*. A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot[J]. Autonomous Robots, 2009, 27(2): 93-103

[10] Kollar T, Roy N. Trajectory optimization using reinforcement learning for map exploration[J]. The International Journal of Robotics Research, 2008, 27(2): 175-196

[11] Yamauchi B. A frontier-based approach for autonomous exploration[C] //Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation. Los Alamitos: IEEE Computer Society Press, 1997: 146-151

[12] Holz D, Basilico N, Amigoni F, *et al*. Evaluating the efficiency of frontier-based exploration strategies[C] //Proceedings of the 41st International Symposium on Robotics and 6th German Conference on Robotics. Berlin: VDE, 2010: 1-8

[13] Ramakrishnan S K, Jayaraman D, Grauman K. An exploration of embodied visual exploration[J]. International Journal of Computer Vision, 2021, 129(5): 1616-1649

[14] Gupta S, Davidson J, Levine S, *et al*. Cognitive mapping and planning for visual navigation[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 7272-7281

[15] Tai L, Liu M. Towards cognitive exploration through deep reinforcement learning for mobile robots[OL]. [2023-01-05]. https://arxiv.org/abs/1610.01733

[16] Mnih V, Kavukcuoglu K, Silver D, *et al*. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533

[17] Chaplot D S, Jiang H, Gupta S, *et al*. Semantic curiosity for active visual learning[C] //Proceedings of the 16th European Conference on Computer Vision. Heidelberg: Springer, 2020: 309-326

[18] Chen T, Gupta S, Gupta A. Learning exploration policies for navigation[OL]. [2023-01-05]. https://openreview.net/forum?id=SyMWn05F7

[19] Schulman J, Wolski F, Dhariwal P, *et al*. Proximal policy optimization algorithms[OL]. [2023-01-05]. https://arxiv.org/abs/1707.06347

[20] He K M, Gkioxari G, Dollár P, *et al*. Mask R-CNN[C] //Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2017: 2980-2988

[21] Hart P E, Nilsson N J, Raphael B. A formal basis for the heuristic determination of minimum cost paths[J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2): 100-107

[22] Wijmans E, Kadian A, Morcos A, *et al*. DD-PPO: learning near-perfect pointgoal navigators from 2.5 billion frames[OL]. [2023-01-05]. https://arxiv.org/abs/1911.00357

[23] Cho K, van Merriënboer B, Bahdanau D, *et al*. On the properties of neural machine translation: encoder-decoder approaches[C] //Proceedings of the 8th Workshop on Syntax, Semantics and Structure in Statistical Translation. Doha: Association for Computational Linguistics, 2014: 103-111

[24] He K M, Zhang X Y, Ren S Q, *et al*. Deep residual learning for image recognition[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2016: 770-778

[25] Savva M, Kadian A, Maksymets O, *et al*. Habitat: a platform for embodied AI research[C] //Proceedings of the IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2019: 9338-9346

[26] Chang A, Dai A, Funkhouser T, *et al*. Matterport3D: learning from RGB-D data in indoor environments[C] //Proceedings of the International Conference on 3D Vision. Los Alamitos: IEEE Computer Society Press, 2017: 667-676

[27] Xia F, Zamir A R, He Z Y, *et al*. Gibson env: real-world perception for embodied agents[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2018: 9068-9079