arXiv:2301.08451v1 [cs.AI] 20 Jan 2023

利用图 Transformer 加速具有有界次优性的多智能体规划

Chenning Yu*, 1, Qingbiao Li*, 2, Sicun Gao[1],Amanda Prorok[2]

摘要—基于冲突的搜索是多智能体寻路中最常用的方法之一。尽管该方法具有完备性和最优性，但扩展性较差。近期研究提出了引入各种启发式方法来加速搜索。然而，这些启发式方法能否应用于非网格问题设置，同时保持其有效性，仍是一个悬而未决的问题。在本研究中，我们发现答案很可能是否定的。为此，我们提出一种基于学习的组件，即图 Transformer，将其作为启发式函数来加速规划。我们证明，所提出的方法具有完备性，并且能够达到任意指定因子的有界次优性。我们在两个具有稠密图的环境中进行了大量实验。结果表明，所提出的图 Transformer 可以在智能体数量相对较少的问题实例上进行训练，并很好地泛化到智能体数量更多的情况，同时性能优于当前最先进的方法。

I. 引言

多智能体路径规划（MAPF）是许多多智能体问题的核心。MAPF 的解是生成无碰撞路径，引导智能体从起始位置前往指定的目标位置。MAPF 在仓库中的物品检索 [1],按需出行服务 [2],监视 [3]以及搜救 [4]等领域具有实际应用。

基于冲突的搜索（CBS）是最受欢迎的 MAPF 规划器之一 [5]。它已被证明是完备且最优的。然而，以最优方式求解 MAPF 是 NP困难 [6], [7]。因此，随着智能体数量增加，搜索空间呈指数级增长，CBS 的可扩展性受到限制。有界次优算法 [8], [9]保证所得解的代价不超过最优解代价的给定常数倍。尽管这些方法在基于网格的多智能体路径规划实例上通常比 CBS 运行得更快，但它们在非网格问题设置中的有效性仍是一个悬而未决的问题；在这类设置中，智能体可以在任意连续域内移动。此外，由于这些启发式方法大多严重依赖碰撞检测来发现冲突，当图较为稠密时，其计算代价可能会相当高。

近年来，基于学习的方法在高效解决多智能体路径寻找（MAPF）任务方面展现出潜力 [10]–[13], ，这些方法将在线计算负担转移到离线学习过程中。然而，我们发现几乎没有此类方法研究基于图的多智能体路径规划。因此，本研究主要探讨基于学习的方法能否以及如何加速非网格环境中的多智能体路径查找规划器，尤其是在稠密图中。

贡献。 我们提出在非网格环境中使用图 Transformer 作为启发式函数，以加速基于冲突的搜索（CBS）。与先前的工作类似 [12], 通过将焦点搜索引入 CBS，我们的框架保证解的完备性和有界次优性。我们的贡献如下：

- 我们提出了一种新颖的架构，即图 Transformer，它利用了多智能体路径寻找问题的内在结构。该架构具有多项理想特性，例如能够处理任意数量的智能体，因此非常适合多智能体路径寻找问题。据我们所知，我们是最早在非网格问题环境下将学习组件引入多智能体路径寻找问题的研究之一。
- 我们设计了一种新颖的训练目标，即对比损失，用于学习对搜索节点进行排序的启发式函数。与 [12],不同，我们的损失函数无需引入上界即可直接优化，因此适用于深度学习。
- 我们通过使用相对较少的智能体进行训练，并在包含更多智能体数量的未见实例上进行测试，展示了模型的泛化能力。结果表明，我们的方法能够显著加速 CBS，而采用手工设计启发式方法的 ECBS [8], 则表现不佳。

相关工作。 主流的多智能体路径寻找规划器主要分为三类：最优规划器、有界次优规划器和无界次优规划器。最优规划器包括 BCP 求解器 [14], [15], 和基于冲突的搜索（CBS） [5], , 以及其后续变体，例如 CBSH [16] 和 CBSH2 [17]。有界次优规划器是另一类研究方向，它们具有更好的可扩展性，同时保证完备性和有界次优性。代表性工作包括 EPEA* [18], 带算子分解的 A* [19],M* [20], ECBS [8],和 EECBS [9]。最后但同样重要的是，无界次优规划器旨在大幅加快规划速度。示例包括优先级规划 [21], ORCA [22], 推送与交换 [23],以及并行推送与交换 [24]。这些方法能够快速找到解，但无法保证解的质量 [25], [26]。

近年来，人们引入基于学习的方法，利用模仿学习 [10], [27] 和强化学习 [13], [28], [29]来高效解决多智能体任务。这些端到端方法通常擅长记忆训练过程中见到的模式，在部署到类似任务时可大幅减少在线计算。 然而，很难确保其完备性。为此，已有多项研究提出训练基于学习的启发式，并用它来引导树搜索 [30]–[33]。为加速 CBS 和 ECBS，Huang 等人 [11], [12]使用支持向量机（SVM）为搜索提供偏置，并通过模仿学习进行训练。我们的工作不仅是首批将机器学习应用于非栅格 问题设置的研究之一，也是首批将深度学习 应用于多智能体路径规划（MAPF），并提供完备性和有界次优性保证的研究之一。

*同等贡献。[1]加利福尼亚大学圣迭戈分校计算机科学与工程系。 < chy010, sicung>@ucsd.edu[2]剑桥大学计算机科学与技术系。 < ql295, asp45>@cam.ac.uk

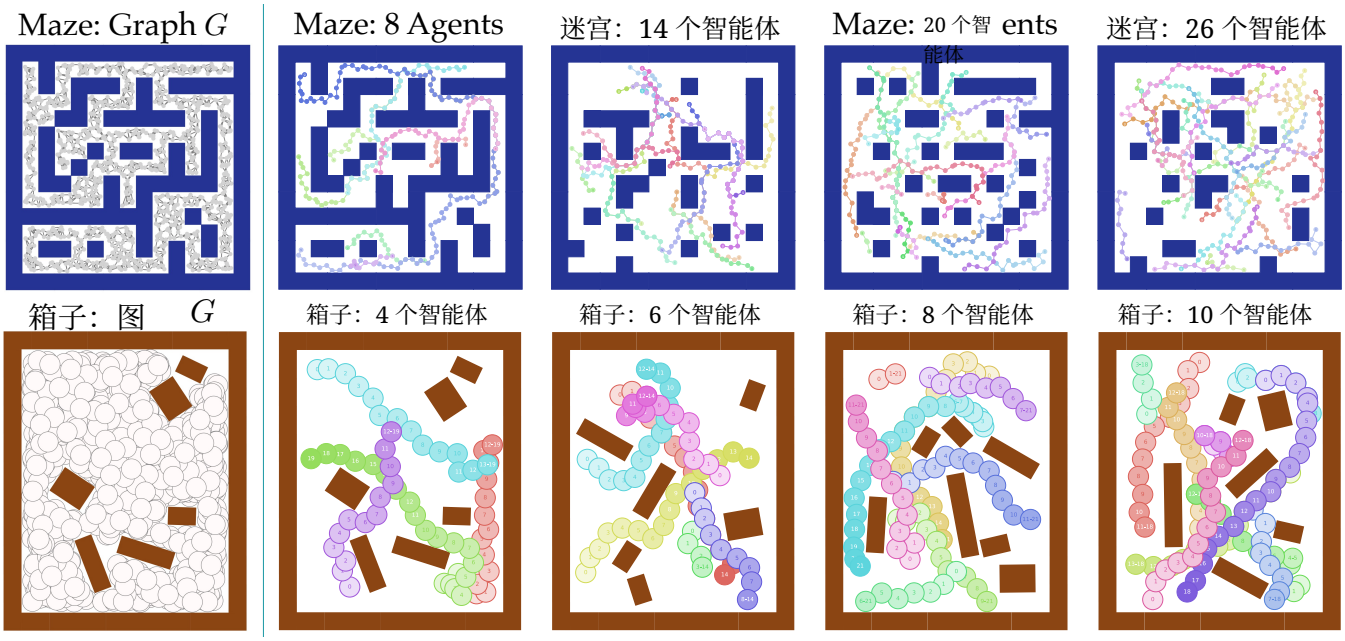


图 1：**左：**基于图的多智能体路径规划实例示例。为构建图，我们从自由空间中随机采样顶点，并用无碰撞边连接它们。**右：**我们的方法能够解决而其他基线方法无法解决的问题实例。不同颜色表示不同智能体的轨迹。同一轨迹中的顶点在对应时间步越晚时颜色越深。

II. 问题表述

我们研究二维连续空间 $\mathcal{C} \subseteq \mathbb{R}^2$中的多智能体路径规划（MAPF）。构型空间 $\mathcal{C}$ 由一组障碍物 $\mathcal{C}_{obs} \subseteq \mathcal{C}$和自由空间 $\mathcal{C}_{free}$组成： $\mathcal{C} \backslash \mathcal{C}_{obs}$。请注意， $\mathcal{C}_{obs}$ 可能与工作空间中呈现的情况不同，因为它还考虑了智能体的几何形状，而智能体未必只是一个质点。

从空间中采样一个随机几何图 $G = \langle V, E \rangle$ 。每个采样顶点 $v \in V$均无碰撞，即$v \in \mathcal{C}_{free}$。若满足以下条件，则有向边 $e \in E$ ：$(v_i \rightarrow v_j)$连接v_i 与 v_j：(i) v_j 是 v_i的邻居之一；(ii) 该边无碰撞，即 $e \subseteq \mathcal{C}_{free}$。邻居集合可以定义为半径为 r 的邻居集合，或 k个最近邻。

假设该图 G上有 M 个智能体。每个智能体 i占据一个区域 $\mathcal{R}(q) \subseteq \mathcal{C}$，该区域与一个顶点 $q \in V$相关联。我们为每个智能体i指定一个起始顶点 s_i 和一个目标顶点 g_i 。我们将智能体 i 的路径记为 σ_i ： $\{v_i^t\}_{t \in [1 \cdots T_i]}$，其中 $T_i \in \mathbb{Z}_{>0}$，且智能体 i 在时间步 t位于顶点 v_i^t。我们将 e_i^t 记为边 $(v_i^t \rightarrow v_i^{t+1})$，它在 1 个时间步内从 v_i^t 移动到v_i^{t+1} 。

问题描述。 我们将一个元组 $(G, \mathcal{S}, \mathcal{G}, \mathcal{C}, \mathcal{R})$ 视为一个多智能体路径寻找问题实例，其中 $\mathcal{S}: \{s_i\}_{i \in [1 \cdots M]}$ 和 $\mathcal{G}$: $\{g_i\}_{i \in [1 \cdots M]}$ 分别是起点和目标顶点。无冲突解 $\{\sigma_i\}_{i \in [1 \cdots M]}$ 应满足以下目标：对于任意时间 t 和一对智能体 i, j [34]:

（端点） $v_i \triangleq s_i \wedge v_i^{T_i} = g_i$
（障碍物） $v_i^t \in \mathcal{C}_{free} \wedge e_i^t \subseteq \mathcal{C}_{free}$
（智能体间） $\mathcal{R}(q_i) \cap \mathcal{R}(q_j) = \emptyset, \ \forall q_i \in e_i^t, \ q_j \in e_j^t$

注意： 如果 $t \geq T_i$，我们假设相应的 v_i^t 等于 $v_i^{T_i}$。这意味着该智能体从时间步 $v_i^{T_i}$ 开始将停留在目标位置。

解的质量。 我们假设每条边都需要 1 个时间步才能通过。解的质量以旅行时间之和（流时间）衡量：$\sum_{i \in [1 \cdots M]} T_i$。

III. 背景：采用有偏启发式的基于冲突的搜索

本节首先介绍基于冲突的搜索（CBS），这是一种最优多智能体规划器 [5]。然后介绍焦点搜索 [35], [36], ，它将有偏启发式纳入 CBS 框架，同时保留有界次优性和完备性保证 [8]。

A. 基于冲突的搜索

基于冲突的搜索（CBS）是一种用于多智能体路径寻找的最优双层树搜索算法。高层规划器旨在解决智能体间冲突，而低层规划器旨在生成最优的个体路径。这里将智能体间冲突表示为 $(i, \ j, \ t, \ v_i^{t-1}, \ v_j^{t-1}, \ v_i^t, \ v_j^t)$，这表示两条边（$v_i^{t-1} \rightarrow v_i^t$）和（$v_j^{t-1} \rightarrow v_j^t$）不满足第 II 节中提到的智能体间目标。

高层规划器维护一棵树，并以最佳优先方式决定扩展哪个搜索节点。为此，每个搜索节点 N 都存储以下信息：

(1) 一组约束 $\mathcal{T}(N)$。约束 $(i, \ v, \ t)$ 表示智能体 i 在时间 t不应移动到图顶点 v 。

(2) 一个解 $\sigma(N)$ ： $\{\sigma_i\}_{i \in [1 \cdots M]}$。该解满足端点目标和障碍物目标，但可能满足也可能不满足智能体间目标。此外，该解应遵守约束 $\mathcal{T}(N)$ ，即 $\forall i, \ \forall v_i^t \in \sigma_i$, $(i, \ v, \ t) \notin \mathcal{T}(N)$ 。

(3) 解的代价 $c(N)$。高层规划器根据此指标决定优先扩展哪个搜索节点。

在高层，CBS 首先创建一个不带约束的根搜索节点，然后不断选择并扩展搜索节点。若某个搜索节点 N^* 是代价最低的叶节点，则会选中该节点。随后，CBS 检查解$\sigma(N^*)$是否存在智能体间冲突。如果没有冲突，$\sigma(N^*)$将作为最终结果返回。否则，CBS

会选择第一个冲突 C：$(i,\ j,\ t,\ v_i^{t-1},\ v_j^{t-1},\ v_i^t,\ v_j^t)$，并将其拆分为两个约束 C_1：$(i,\ v_i^t,\ t)$ 和 C_2：$(j,\ v_j^t,\ t)$。随后生成两个子搜索节点 N_1 和 N_2，其约束分别为 $\forall i=1、2,\ \mathcal{T}(N_i)=\mathcal{T}(N^*)\cup\{C_i\}$。然后，每个子搜索节点都会调用一个最优低层规划器（例如 A* [37],），为每个受影响的智能体重新规划路径，并记录相应的解和代价。CBS 能保证完备性和最优性，因为高层和低层规划器都在执行最佳优先搜索 [5]。

B. 使用焦点搜索纳入有偏启发式

CBS 是一种最优规划器，但即使在基于网格的问题设置中，其扩展性也不佳。为提高扩展性，先前的工作引入了焦点搜索 [35], [36]，例如有界冲突搜索（BCBS）和增强型冲突搜索（ECBS） [8]。下面我们介绍 BCBS 的一个简化版本。

我们在算法1中给出了焦点搜索的伪代码。焦点搜索将焦点集合引入 CBS 框架。焦点集合（$Focal$）会在冲突搜索树中保留一定比例的叶搜索节点（即 $Open$）。我们将 LB 记为叶搜索节点中的最低解代价。所有满足近似最优解质量要求$c \le w\cdot LB$ 的叶搜索节点都会被加入 $Focal$。这个新的 CBS 将从 $Focal$ 中选择一个搜索节点进行扩展，而不是从$Open$中选择。与原始 CBS 相比，新算法同样会在 $Focal$上执行最佳优先搜索，但搜索优先级从解的代价变为一个新的启发式函数 ψ。通常，ψ 是一个手工设计的函数，以解为输入，并输出一个倾向于选择冲突较少的解的值。我们则使用基于**图 Transformer**的学习型启发式函数。

算法1 采用有偏启发式的 CBS [8], [12]

输入： 一个多智能体路径查找实例和次优因子 w
输入： 启发式函数 ψ （例如，图 Transformer）
生成根搜索节点 R，并为其提供一个初始解
初始化 Open 列表 $Open \leftarrow \{R\}$
$LB \leftarrow c(R)$，并初始化焦点集列表 $Focal \leftarrow \{R\}$
当 $Open$非空
  $N^* \leftarrow$使$_{N\in Focal}\ \psi(\sigma(N))$最小的参数
  $C \leftarrow$ $\sigma(N^*)$中的第一个冲突
  如果 C 不存在
    返回解 $\sigma(N^*)$
  从 $Open$和 $Focal$中移除 N^*
  如果 $\min_{N\in Open} c(N) > LB$
    $LB =$ 分钟$_{N\in Open}\ c\ (N)$
    $Focal = \{N \in Open:\ c(N) \le w\cdot LB\}$
  从节点 N^*生成两个子节点 N_1 和 N_2
  将 C_i 添加到 $\mathcal{T}(N_i)$ 中，对于 $i=1,\ 2$
  调用低层规划器以获取 $\sigma(N_i)$，对于 $i=1,\ 2$
  将 N_i 添加到 $Open$，对于 $i=1,\ 2$
  将 N_i 添加到 $Focal$，如果 $c(N_i) \le w\cdot LB$，对于 $i=1,\ 2$
返回无解

命题。 如 *[8]* 所述，算法*1*是完备的，且有界次优因子为 $w \ge 1$。

假设该问题有可行解，但在给予充足的时间预算后，算法1仍未找到解。那么，对于任意一个具有可行解的搜索节点 N，必然存在一个祖先搜索节点 N^p，它已加入 $Open$，但尚未展开。假设 N^{p*} 是这些未展开祖先中代价最低的搜索节点。由于 N^{p*} 尚未展开，因此不会被 $Focal$ 选中。因此，要么 (i) N^{p*} 不在 $Focal$ 中，要么 (ii) N^{p*} 在 $Focal$ 中但未被选中。(i) 不可能，因为代价低于 $\frac{1}{w}\cdot c(N^{p*})$ 的解不可能有无穷多个。(ii) 也不可能，因为代价低于或等于 $w\cdot c(N^{p*})$ 的解不可能有无穷多个。因此，N^{p*} 最终会被选中并展开。由此，我们通过反证证明了该算法是完备的。焦点搜索绝不会展开代价高于最优解 w 倍的搜索节点；因此，它是有界次优的，次优因子为 w。

请注意，此处描述的焦点搜索是 BCBS [8]，（即 BCBS (w,1)）的一个特例，因为焦点搜索仅应用于高层规划器。在我们的基于图的问题设置中，将焦点搜索应用于低层规划器并无显著改进。相反，如果将焦点搜索引入低层规划器，则会消耗大量计算资源来进行边的碰撞检测，这并不能提升整体性能。有关详情，请参阅第V-B节中的问题4。

IV. 图变换器作为启发式函数

本节介绍图 Transformer 的架构，以及如何对其进行训练，使其表示一种能够加速 CBS 的启发式函数。

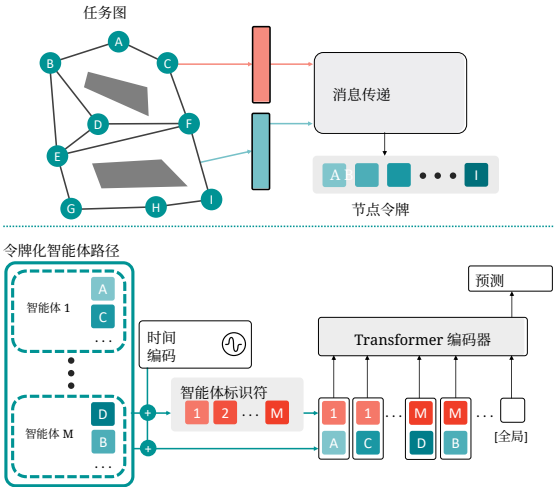


图2：所提出的图 Transformer 架构。它具有若干专为处理多智能体路径寻找输入而设计的理想属性。更多详情请参见第IV-A节。

A. 网络架构

图 Transformer 的输入 ϕ 是一个图 G和一个解 $\sigma = \{\sigma_i\}_{i\in[1\cdots M]}$。输出 $\phi(G,\sigma)$ 预测一个标量值，作为启发式值。该预测值与当前搜索节点产生具有可行解的后代搜索节点的可能性相关。举例来说，与最终无法减少冲突的节点相比，能够导向可行解的有希望节点应具有较低的 $\phi(G,\sigma)$ 值。

图 Transformer 分为两个阶段：图标记化和注意力聚合。下面分别介绍这两个阶段。

阶段 1：图标记化。图标记化使用图神经网络（GNN）将每个图顶点转换为嵌入。这里我们采用消息传递神经网络 [38]（MPNN）作为 GNN 架构。MPNN 的输入是一个图 $G = \langle V, E \rangle$，其中每个图顶点 $v_i \in V$ 的特征 d_i^v 是其对应的二维位置，而每条边 $e_l = (v_i \to v_j)$ 的特征 d_l^e 是 v_j 相对于 v_i 的位置。经过两个线性层f_x 和 f_y，顶点和边首先分别编码为 x和 y，使用 $\forall v_i \in V,\ x_i = f_x(d_i^v)$；$\forall e_l \in E,\ y_l = f_y(d_l^e)$。随后，MPNN 使用三个多层感知机 $\{f_k\}_{k\in[1,2,3]}$，按如下方式更新每个图顶点 $v_i \in V$ 的信息：

$$x_i \leftarrow x_i + \max\{f_k(x_i, x_j, y_l)\}, \forall e_l : (v_i \to v_j) \in E. \quad (1)$$

使用多层感知机 f_1更新完所有 x_i 后，消息传递神经网络会继续使用多层感知机 f_2 更新 x_i，依此类推。max 表示沿特征维度进行最大池化。我们使用最大池化来处理元素数量任意的集合，同时确保鲁棒性 [39]。由于它对这些元素的排列不变，因此此处的消息传递神经网络不仅可以处理顶点和边数量任意的图，而且在构造上也具有排列不变性。

阶段 2：注意力聚合。在第一阶段为每个图顶点 v_i 计算出令牌 x_i 后，我们使用 Transformer [40]对智能体间交互进行建模。首先将每个智能体 σ_i 的路径标记为令牌序列 $\rho_i = \{x_j, \forall v_j \in \sigma_i\}$。为了将时间信息注入这些经过标记的解中，我们引入时间编码 [31]。该方法与 [40], [41] 类似（在其设置中用作位置编码）。我们将 $\rho_i^t \in \mathbb{R}^D$ 表示为智能体 i 在时间步 t的顶点令牌。对于每个令牌 ρ_i^t，我们将其与时间编码 $\rho_i^t \leftarrow \rho_i^t + TE(t) \in \mathbb{R}^D$逐元素相加。$TE(t)$ 的第 $2k$ 和第2k+1 个维度如下：

$$TE(t)_{2k} = \sin(t/10000^{2k/D}), \quad (2)$$

$$TE(t)_{2k+1} = \cos(t/10000^{2k/D}). \quad (3)$$

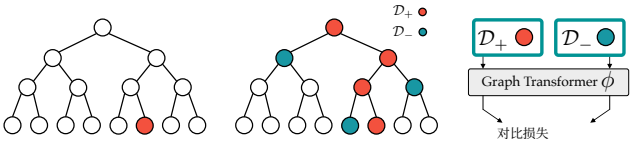
按照常见做法，使用超参数 10000 [40]。为了让模型能够识别每个令牌 ρ_i^t 所属的智能体，我们将智能体标识符 τ_i 与每个令牌 $\rho_i^t \leftarrow \rho_i^t || \tau_i$拼接起来，类似于 [42]。对于每个智能体 i，智能体标识符 τ_i 通过对其所有顶点令牌执行最大池化计算得出：$\tau_i = \max\{\rho_i^t\},\ \forall t \in [1 \cdots T_i]$。然后，所有智能体的解中的全部令牌$\{\rho_i^t :\ \forall i \in [1 \cdots M],\ \forall t \in [1 \cdots T_i]\}$ 将作为输入馈入 Transformer 编码器。按照常见做法 [43], [44]，我们在输入中附加一个额外的可训练令牌 $global$，以进行全局预测。Transformer

编码器为每个输入 token 预测一个输出，我们将可训练的 $global$ token 的输出用作 ζ。通过线性层 $f_\phi : \mathbb{R}^D \to \mathbb{R}$，最终输出计算为$\phi(G,\ \sigma) = f_\phi(\zeta)$。

这里我们使用 Transformer 编码器，因为它使图 Transformer 能够接受数量可变的 token 并对其依赖关系建模，同时保持对 token 的排列的不变性。关于 Transformer 编码器的更多详情，请读者参阅 [40]。

图 Transformer 的特性。从构造上看，图 Transformer 能够处理顶点数和边数可变的输入图、智能体总数可变的情况，以及长度可变的输入解。此外，它还能感知时间信息和智能体间交互。其输出对于图顶点和智能体的排列顺序均具有置换不变性。

B. 训练图变换器



a. CBS 找到一个可行解　b. 样本标注　c. 对比训练

图 3：训练框架。我们使用监督式对比损失。标签由 CBS 搜索树生成。

数据生成。给定一个多智能体路径查找实例，我们首先使用 CBS 生成可行解。如果 CBS 未能求解该实例，则不会收集任何数据。如果求解成功，我们便开始从其搜索树中收集正样本和负样本。如果一个样本本身或其某个后代搜索节点包含该可行解，则将其作为正样本收集到数据集 $\mathcal{D}_+$ 中。如果一个搜索节点是正样本的兄弟搜索节点，则将其作为负样本收集到数据集 $\mathcal{D}_-$ 中。我们记录每个样本的图 G 和解 σ。此外，我们记录值 d，作为其在搜索树中的相应深度。

监督式对比学习。模型的目标是学习样本的排序。也就是说，给定来自同一个多智能体路径寻找图 G 的任意一对正样本 (G, σ_+, d_+) 和负样本 (G, σ_-, d_-)，我们学习以下排序：

$$\phi(G, \sigma_+) < \phi(G, \sigma_-),\ \text{if } d_+ \geq d_-.$$

我们在此阐释其直观含义。设想这种排序已被完美学得，且 $w = \infty$，这意味着所有叶搜索节点都将位于 $Focal$ 中。假设在某个时刻，$Focal$ 包含 1 个正样本 p_+。$Focal$ 可能包含负样本，也可能不包含。如果存在负样本，那么它们的深度均不大于 $d(p_+)$。将此条件定义为循环不变式。随后，根据排序，p_+ 将首先被选中并展开。如果 p_+ 是可行解，则算法终止。否则，$Focal$ 将包含一个正样本 p'_+ 和若干负样本，且所有负样本的深度均不大于 $d(p'_+)$。因此，循环不变式仍然成立。对于基本情况，循环不变式也成立，此时 $Focal$ 仅包含根搜索节点。因此，通过学习这样的排序，我们促使算法1优先展开正样本，并尽可能少地展开负样本，从而节省大量计算并显著加速 CBS。

我们使用监督式对比学习来训练图 Transformer ϕ，使其将正样本排在负样本之前。给定任意一对正样本和负样本，$p_+ : (G_+,\ \sigma_+,\ d_+) \in \mathcal{D}_+$，$p_- : (G_-,\ \sigma_-,\ d_-) \in \mathcal{D}_-$，将这对样本定义为有效样本对的条件为：$\mathbb{I}(p_+,\ p_-) : (G_+ = G_-) \wedge (d_+ \geq d_-)$。对于超参数 $\gamma = 0.1$，我们定义$\delta(x) : \max(0,\ \gamma + x)$。我们的目标是最小化对比损失，如下所示：

$$\frac{1}{L} \sum_{\substack{p_+ \in \mathcal{D}_+ \\ p_- \in \mathcal{D}_-}} \delta(\phi(G_+, \sigma_+) - \phi(G_-, \sigma_-)) \cdot \mathbb{I}(p_+, p_-), \quad (4)$$

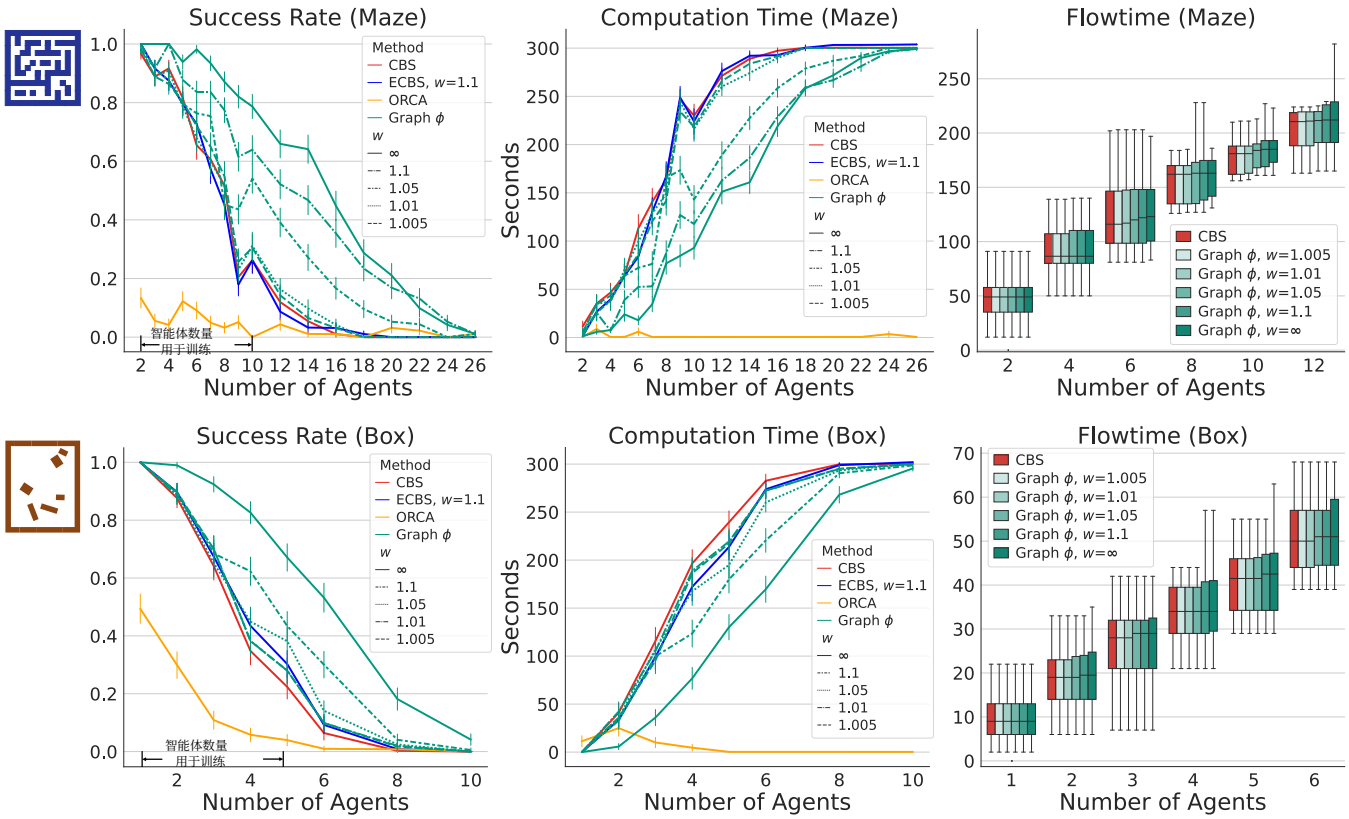


图4：在5分钟运行时间限制内，成功率、计算时间和流时间随智能体数量变化的情况。对于每种智能体数量设置，结果均取100个测试实例的平均值。我们使用 $w \in [1.005, 1.01, 1.05, 1.1, \infty]$评估所提出的方法，并将其性能与CBS、ECBS（$w = 1.1$）和ORCA进行比较。尽管训练时使用的智能体数量相对较少，结果表明，我们的方法具有良好的泛化能力，并显著优于基线方法。

其中， $L = \sum_{p_- \in \mathcal{D}_-}^{p_+ \in \mathcal{D}_+} \mathbb{I}(p_+, p_-)$ 是有效样本对的总数， $\mathcal{D}^+$, $\mathcal{D}^-$ 分别是正样本和负样本的数据集。

图 Transformer ϕ 的训练达到收敛后，我们可以将其部署为算法1中的启发式函数 ψ。然而，在实践中我们发现，该模型会对焦点集合中的多个搜索节点预测出相对较低的值，这表明它们都可能导向可行解。为打破平局，我们改为将ψ表示为深度 d 与 ϕ的组合，即 $\psi = \langle -d, \phi \rangle$。这意味着算法会优先选择深度较大的搜索节点；如果存在多个深度最大的搜索节点，则优先选择 ϕ较低的搜索节点。这种设计使算法在存在多个有希望的节点时能够始终如一地做出决策。若无进一步说明，我们将采用此类启发式的方法记作**图** ϕ。

V. 实验

A. 主要实验

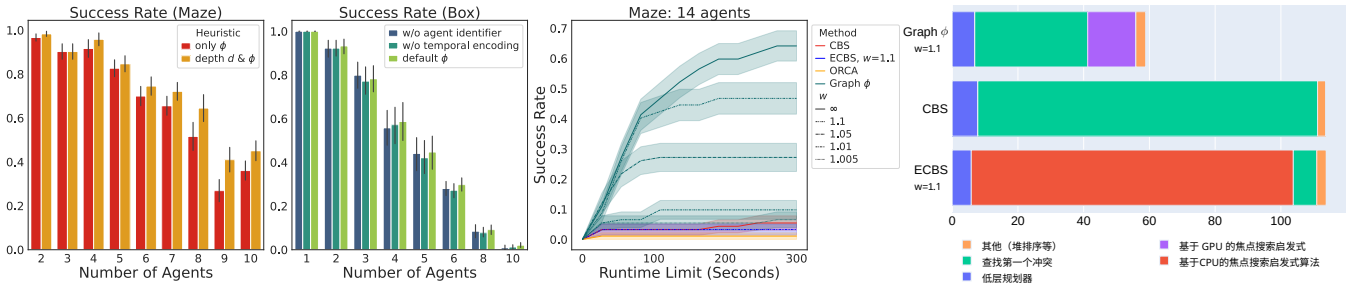
实验设置。 我们设计了两种环境用于评估：迷宫和箱子（见图 1）。每种环境均包含 2700 个多智能体路径查找实例，训练样本由 CBS 生成。测试时，每种智能体数量设置下有 100 个多智能体测试实例。我们为每个迷宫实例生成一张随机地图，并为每个箱子实例生成一组随机障碍物。每个实例的图以及起点和目标顶点也均为随机生成。

迷宫环境的训练实例包含 2 至 10 个智能体，箱子环境的训练实例包含 1 至 5 个智能体。迷宫环境的测试实例包含 2 至 26 个智能体（2–10、12、14、16、18、20、22、24、26），箱子环境的测试实例包含 1 至 10 个智能体（1–5、6、8、10）。我们确保测试实例未出现在训练集中。所有实验均使用一台 12 核、3.2 GHz 的 i7-8700 中央处理器和 4 块 Nvidia GTX 1080Ti 显卡进行。我们测试了 $w \in [1.005, 1.01, 1.05, 1.1, \infty]$，以评估我们的方法。对于所有方法，我们按照惯例 [12]，将运行时间限制设为 5 分钟。

基线方法。 我们将我们的方法与 3 种基线方法进行比较：(i) 基于冲突的搜索。(ii) 增强型基于冲突的搜索 (ECBS)：CBS [8] 的一种有界次优版本。它在高层和低层规划器中都应用焦点搜索，并使用手工设计的启发式函数。我们选择其 $w = 1.1$。(iii) ORCA [22]：一种反应式碰撞规避算法，在低密度环境中表现良好。

我们发现，几乎没有开源实现能够将 CBS 和 ECBS 直接应用于非网格问题。因此，我们使用 Python 从头实现了所有树搜索方法（CBS、ECBS 和我们的方法）。在实现 CBS、ECBS 和我们的方法时，我们基本采用了相同的框架。为确保比较公平，我们严格遵循 C++实现 [1] 和原始论文 [5], [8], ，并使用贝叶斯超参数搜索来优化 ECBS 的 w，从而对这三种方法进行充分优化。

[1]https://github.com/whoenig/libMultiRobotPlanning/



(a) 启发式 (b) 网络架构 (c) 运行时间限制 (d) 时间分析（秒）

图5：我们进行了4项不同的消融实验，以系统地评估所提出的方法。更多详情请参见第V-B节。

评估指标。我们的评估包括 3 项指标：1）成功率，即成功实例数占测试实例总数的比例。如果所有智能体都在超时前无碰撞地到达目标，则该实例视为成功。2）计算时间，每个实例的平均计算时间，包括失败的实例。[2] 3）流时间，所有智能体行进时间的总和。我们仅根据 CBS 和我们的方法都成功的实例，对比两者的流时间，以展示我们的方法所具备的有界次优性保证。

总体性能。我们在图4中展示了总体性能。无论是在迷宫还是箱子环境中，我们的方法都显著优于三种基线方法。它所需的计算时间少得多，成功率也显著更高。此外，尽管训练时使用的智能体数量相对较少，我们的方法仍能很好地泛化到更多智能体。例如，在迷宫环境中，使用2至10个智能体训练的网络最多可泛化至26个智能体；而在箱子环境中，使用1至5个智能体训练的策略最多可泛化至10个智能体。具体而言，在迷宫环境中，当智能体数量为14、18、22和26时，我们的方法在 $w = 1.1$ 下的成功率分别为47%、23%、10%、1%；在箱子环境中，当智能体数量为4、6、8和10时，成功率分别为62%、30%、4%0、5%。另一方面，CBS在超时前无法求解包含18个智能体的迷宫环境和包含10个智能体的箱子环境。类似地，对于迷宫环境中智能体数量达到20个以及箱子环境中智能体数量达到10个的所有实例，ECBS都开始出现求解失败。此外，由于理论上已证明我们算法的有界次优性，因此我们的方法所得解的质量（流时间）非常接近最优解，这并不令人意外。最后，ORCA在高密度环境中很容易失效，这一表现与先前的研究一致 [45]。

B. 消融研究

在本节中，我们探讨四个问题：(

a) Is i　　ing depth inf　格式　　ion i

到启发式中是否有利于性能？图5(a)展示了在迷宫中针对2至10个智能体的测试中，仅使用 ϕ的启发式与使用深度 d 和 ϕ的启发式之间的比较。结果表明，引入深度信息是有效的。随着智能体数量增加，性能提升更加显著，这验证了我们的选择。

（b）时间编码和智能体标识符是否能提升性能？图5(b)展示了在箱子中使用和不使用时间编码时的性能

[2]由于CBS和ECBS调用不同的低层规划器（A*和A*-ϵ），我们认为计算时间是评估效率最公平的指标，而不是例如统计扩展节点的数量。

ls.

编码和智能体标识符的结果。结果表明，智能体标识符和时间编码能够提升成功率。相较于非默认设置，引入智能体标识符和时间编码后的平均提升幅度为 16±33%。

（c）运行时间限制会如何影响性能？我们以迷宫中包含 14 个智能体的测试为例，将运行时间限制设为 25 秒至 300 秒，间隔为 25 秒。图5（c）显示，无论运行时间限制如何变化，我们的方法都优于基线方法。当限制为 300 秒时，我们的方法成功率达到 64%（$w = \infty$），而 CBS、ECBS 和 ORCA 的成功率分别仅为 5%、3% 和 1%。

（d）规划器各模块的时间分析。最后，我们希望回答这样一个问题：与传统的基于网格的设置相比，将 ECBS 应用于非网格多智能体路径查找实例时，其性能为何会大幅下降？我们对各方法进行分析，并对迷宫中包含 2 至 10 个智能体的测试结果取平均值。

图5（d）显示，ECBS 在焦点启发式计算上耗费了大量计算资源。这种情况并不意外，因为对于稠密图，启发式需要通过检查边上的碰撞来计算。这种碰撞检测通常是规划的主要瓶颈，而且一般来说本身就是 NP 困难问题 [46], [47]。与此同时，我们的方法使用学习函数计算焦点启发式（图形处理器部分），其计算代价仅为 ECBS 的 15%。与 CBS 和 ECBS 相比，我们的方法所需的总计算时间仅为其 50%。

VI. 结论与未来工作

我们提出了图 Transformer，作为一种启发式函数，通过引导树搜索朝着可能存在可行解的有希望节点前进，来加速 CBS。在显著提升速度的同时，我们的方法保证了可证明的完备性和有界次优性。我们表明，在具有稠密运动图的双连续环境中，我们的方法优于三种经典的多智能体路径寻找基线方法（CBS、ECBS 和 ORCA），并且能够很好地泛化到智能体数量更多的未见测试中。未来的工作包括开展真实世界实验，以及改进该方法以支持数量提升 10 至 100 倍的智能体。

致谢。本材料基于以下项目的支持：ARL DCIST CRA W911NF-17-2-0181、欧洲研究理事会（ERC）项目949940（gAIa）、美国空军与DARPA根据合同编号FA8750-18-C-0092提供的支持、AFOSR YIP FA9550-19-1-0041、NSF Career CCF 2047034、亚马逊研究奖，以及Amazon.com Inc.提供的捐赠。

参考文献

[1] J. Enright 和 P. R. Wurman，“移动履约系统中的优化与协同自主性”，载于AAAI自主移动机器人自动行动规划研讨会，2011年，第33–38页。 [2] A. Prorok 和 V. Kumar，“按需出行系统中的保护隐私的车辆分配”，载于IEEE/RSJ智能机器人与系统国际会议，2017年，第1869–1876页。 [3] B. Grocholsky、J. Keller、V. Kumar 和 G. Pappas，“空地协同监视”，《IEEE机器人与自动化杂志》，第13卷，第3期，第16–25页，2006年。 [4] V. Kumar 和 N. Michael，“自主微型飞行器的机遇与挑战”，载于《机器人学研究》，施普林格，2017年，第41–58页。 [5] G. Sharon、R. Stern、A. Felner 和 N. R. Sturtevant，“用于最优多智能体寻路的基于冲突的搜索”，《人工智能》，第219卷，第40–66页，2015年。 [6] J. Banfi、N. Basilico 和 F. Amigoni，“带孔洞二维网格图上的时间最优多机器人路径规划的难处理性”，《IEEE机器人与自动化快报》，第2卷，第4期，第1941–1947页，2017年。 [7] J. Yu 和 S. M. LaValle，“图上多机器人的最优路径规划”，载于2013年IEEE机器人与自动化国际会议。IEEE，2013年，第3612–3617页。 [8] M. Barer、G. Sharon、R. Stern 和 A. Felner，“多智能体寻路问题的基于冲突的搜索算法的次优变体”，载于欧洲人工智能会议，2014年，第961–962页。[9] J. Li、W. Ruml 和 S. Koenig，“EECBS：一种用于多智能体寻路的有界次优搜索”，载于AAAI人工智能会议论文集，第35卷，第14期，2021年，第12 353–12 362页。 [10] Q. Li、F. Gama、A. Ribeiro 和 A. Prorok，“用于分布式多机器人路径规划的图神经网络”，载于IEEE/RSJ智能机器人与系统国际会议，2020年，第11 785–11 792页。 [11] T. Huang、S. Koenig 和 B. Dilkina，“通过基于冲突的搜索学习解决多智能体寻路中的冲突”，载于AAAI人工智能会议论文集，第35卷，第13期，2021年，第11 246–11 253页。 [12] T. Huang、B. Dilkina 和 S. Koenig，“学习用于多智能体寻路的有界次优基于冲突搜索中的节点选择策略”，载于自主智能体与多智能体系统国际联合会议，2021年。 [13] G. Sartoretti、J. Kerr、Y. Shi、G. Wagner、T. Kumar、S. Koenig 和 H. Choset，“PRIMAL：通过强化学习与模仿学习进行多智能体寻路”，《IEEE机器人与自动化快报》，第4卷，第3期，第2378–2385页，2019年。 [14] E. Lam 和 P. Le Bodic，“ 多智能体寻路分支定界切平面算法中的新有效不等式”，载于自动规划与调度国际会议论文集，第30卷，2020年，第184–192页。 [15] E. Lam、P. Le Bodic、D. Harabor 和 P. J. Stuckey，“用于多智能体寻路的分支定界切平面算法”，《计算机与运筹研究》，第144卷，文章编号105809，2022年。 [16] A. Felner、J. Li、E. Boyarski、H. Ma、L. Cohen、T. S. Kumar 和 S. Koenig，“为多智能体寻路的基于冲突的搜索添加启发式方法”，载于自动规划与调度国际会议论文集，第28卷，2018年，第83–87页。 [17] J. Li、A. Felner、E. Boyarski、H. Ma 和 S. Koenig，“通过基于冲突的搜索改进多智能体寻路的启发式方法”，载于IJCAI，第2019卷，2019年，第442–449页。 [18] A. Felner、M. Goldenberg、G. Sharon、R. Stern、T. Beja、N. R. Sturtevant、J. Schaeffer 和 R. Holte，“带选择性节点生成的部分扩展A*”，载于第二十六届AAAI人工智能会议论文集，2012年7月22日至26日，加拿大安大略省多伦多，J. Hoffmann 和 B. Selman 编。AAAI出版社，2012年。 [在线]。获取地址：http://www.aaai.org/ocs/index.php/AAAI/ AAAI 12 5036 /paper/view/ T. S. Standley，“为协同寻路问题寻找最优解”，载于第二十四届AAAI人工智能会议论文集，AAAI 2010，美国佐治亚州亚特兰大，2010年7月11日至15日，M. Fox 和 D. Poole 编。AAAI出版社，2010年。 [在线]。获取地址：http://www.aaai.org/ocs/index.php/AAAI/ AAAI 10 1926 /paper/view/ G. Wagner 和 H. Choset，“M*：一种具有性能界的完备多机器人路径规划算法”，载于2011年IEEE/RSJ智能机器人与系统国际会议（IROS 2011），美国加利福尼亚州旧金山，2011年9月25日至30日。IEEE，2011年，第3260–3267页。 [在线]。获取地址：https://doi.org/10.1109/IROS.2011.6095022 [21] J. van den Berg、M. Lin 和 D. Manocha，“多机器人操作的互惠速度障碍”，载于IEEE/RSJ智能机器人与系统国际会议，2005年，页码 430–435。[22] J. Van den Berg、M. Lin 和 D. Manocha，“用于实时多智能体导航的互惠速度障碍”，载于 IEEE机器人与自动化国际会议，2008年，页码 1928–1935。[23] R. J. Luna 和 K. E. Bekris，“推送与交换：具有完备性保证的快速协作路径搜索”，载于第二十二届人工智能国际联合会议，2011年。[24] Q. Sajid、R. Luna 和 K. Bekris，“通过同步执行单智能体基元实现多智能体寻路”，载于组合搜索国际研讨会，第 3 卷，第 1 期，2012年。[25] M. ˇCáp、P. Novák、A. Kleiner 和 M. Seleck`y，“多移动机器人轨迹协调的优先级规划算法”，IEEE自动化科学与工程汇刊，第 12 卷，第 3 期，页码 835–849，2015年。[26] J. P. Van Den Berg 和 M. H. Overmars，“动态环境中基于路线图的运动规划”，IEEE机器人学汇刊，第 21 卷，第 5 期，页码 885–897，2005年。[27] Q. Li、W. Lin、Z. Liu 和 A. Prorok，“用于大规模多机器人路径规划的消息感知图注意力网络”，IEEE机器人与自动化快报，2021年。[28] B. Wang、Z. Liu、Q. Li 和 A. Prorok，“通过全局引导强化学习实现动态环境中的移动机器人路径规划”，IEEE机器人与自动化快报，第 5 卷，第 4 期，页码 6932–6939，2020年。[29] C. Yu、H. Yu 和 S. Gao，“利用图神经网络学习多智能体导航的控制可采纳性模型”，2022年。 [30] [ ] 2210 09378 在线。获取地址：https://arxiv.org/abs/。E. B. Khalil、B. Dilkina、G. L. Nemhauser、S. Ahmed 和 Y. Shao，“学习在树搜索中运行启发式方法”，载于 IJCAI，2017年，页码 659–666。[31] R. Zhang、C. Yu、J. Chen、C. Fan 和 S. Gao，“利用图神经网络和时间编码，在动态环境中进行基于学习的运动规划”，CoRR，卷号 abs/2210.08408，2022年。 [在线]。获取地址：https://doi.org/ 10 48550 2210 08408 。/arXiv.。 [32] J. Song、R. Lanka、A. Zhao、A. Bhatnagar、Y. Yue 和 M. Ono，“通过回顾式模仿学习搜索”，arXiv 预印本 arXiv:1804.00846，2018年。[33] C. Yu 和 S. Gao，“利用图神经网络减少基于采样的运动规划中的碰撞检测”，神经信息处理系统进展，第 34 卷，页码 4274–4289，2021年。[34] K. Okumura、R. Yonetani、M. Nishimura 和 A. Kanezaki，“CTRMS：学习构建用于连续空间多智能体路径规划的协作定时路线图”，arXiv 预印本 arXiv:2201 09467，2022年。。[35] J. Pearl 和 J. H. Kim，“半可采纳启发式方法研究”，IEEE模式分析与机器智能汇刊，第 4 期，页码 392–399，1982年。[36] M. Ghallab 和 D. G. Allard，“A：一种高效的近似可采纳启发式搜索算法”，载于人工智能国际联合会议论文集。Citeseer，1983年，页码 789–791。[37] P. E. Hart、N. J. Nilsson 和 B. Raphael，“最小代价路径启发式确定的形式化基础”，IEEE系统科学与控制论汇刊，第 4 卷，第 2 期，页码 100–107，1968年。[38] J. Gilmer、S. S. Schoenholz、P. F. Riley、O. Vinyals 和 G. E. Dahl，“用于量子化学的神经消息传递”，载于国际机器学习会议，2017年，页码 1263–1272。[39] C. R. Qi、H. Su、K. Mo 和 L. J. Guibas，“PointNet：面向点集的三维分类与分割深度学习”，载于 2017 IEEE计算机视觉与模式识别会议，CVPR 2017，美国夏威夷州檀香山，2017年7月21–26日。IEEE计算机学会，2017年，页码 77 85 [ ] 10 1109 2017 16 –。 [40] 在线。获取地址：https://doi.org/。/CVPR.。A. Vaswani、N. Shazeer、N. Parmar、J. Uszkoreit、L. Jones、A. N. Gomez、Ł. Kaiser 和 I. Polosukhin，“注意力就是一切”，载于神经信息处理系统进展，2017年，页码 5998–6008。[41] B. Mildenhall、P. P. Srinivasan、M. Tancik、J. T. Barron、R. Ramamoor-thi 和 R. Ng，“NeRF：将场景表示为神经辐射场以合成视图”，ACM通讯，第 65 卷，第 1 期，页码 99–106，2021年。[42] J. Kim、T. D. Nguyen、S. Min、S. Cho、M. Lee、H. Lee 和 S. Hong，“纯 Transformer 是强大的图学习器”，arXiv 预印本 arXiv:2207.02505，2022年。[43] J. Devlin、M.-W. Chang、K. Lee 和 K. Toutanova，“BERT：用于语言理解的深度双向Transformer预训练”，arXiv 预印本 arXiv:1810.04805，2018年。[44] A. Dosovitskiy、L. Beyer、A. Kolesnikov、D. Weissenborn、X. Zhai、T. Unterthiner、M. Dehghani、M. Minderer、G. Heigold、S. Gelly 等，“一张图像胜过 16×16 个词：用于大规模图像识别的 Transformer”，arXiv 预印本 arXiv:2010.11929，2020年。

[45] S. H. Arul、A. J. Sathyamoorthy、S. Patel、M. W. Otte、H. Xu、M. C. Lin 和 D. Manocha，“Lswarm：复杂城市场景中具有覆盖约束的大规模群体高效碰撞规避”，*IEEE*机器人与自动化快报，第 4 卷，第 4 期，页码 3940–3947，2019年。 [ ] 10 1109 2019 2929981 在线。获取地址：https://doi.org/。/LRA.。B. Chazelle，“多面体的凸分割：下界与最坏情况最优算法”，*SIAM*计算学报，第 13 卷，第 3 期，页码 488–507，1984年。
[46]
[47] P. Jiménez、F. Thomas 和 C. Torras，“用于运动规划的碰撞检测算法”，机器人运动规划与控制，页码 305–343，1998年。