

# 그래프 기반 상태 표현을 활용한 작업 계획 알고리즘 개발

## Task Planning Algorithm with Graph-based State Representation

변성완<sup>1</sup>, 오윤선<sup>†</sup>

Seongwan Byeon<sup>1</sup>, Yoonseon Oh<sup>†</sup>

**Abstract:** The ability to understand given environments and plan a sequence of actions leading to goal state is crucial for personal service robots. With recent advancements in deep learning, numerous studies have proposed methods for state representation in planning. However, previous works lack explicit information about relationships between objects when the state observation is converted to a single visual embedding containing all state information. In this paper, we introduce graph-based state representation that incorporates both object and relationship features. To leverage these advantages in addressing the task planning problem, we propose a Graph Neural Network (GNN)-based subgoal prediction model. This model can extract rich information about object and their interconnected relationships from given state graph. Moreover, a search-based algorithm is integrated with pre-trained subgoal prediction model and state transition module to explore diverse states and find proper sequence of subgoals. The proposed method is trained with synthetic task dataset collected in simulation environment, demonstrating a higher success rate with fewer additional searches compared to baseline methods.

**Keywords:** Task Planning, Artificial Intelligence, Graph Neural Network, Search Algorithm

### 1. 서론

인구 고령화 및 개인화가 심화되면서 개인의 일상적인 작업을 보조하는 서비스로봇에 대한 수요가 증가하고 있다<sup>[1]</sup>. 인간이 일상적으로 수행하는 대부분 작업은 주어진 목표 조건을 달성하기 위해 특정 물체를 조작하는 행동의 연속으로 이루어져 있다. 예를 들어 책상 위에 박스들이 쌓여 있는 상황에서 아래에 놓인 박스를 집기 위해서는 먼저 위에 올려진 박스를 집어서 옆으로 옮기는 행동을 수행해야 한다. 이러한 작업을 로봇이 수행하기 위해서는 주어진 목표와 환경의 현재 상태를

이해하고, 이를 바탕으로 필요한 행동을 계획하는 작업 계획 (Task Planning) 능력이 필요하다<sup>[2]</sup>.

작업 계획은 크게 환경 인식과 세부 목표(Subgoal) 예측의 두 단계로 이루어진다. 환경 인식 단계에서는 작업 환경에서 초기 및 목표 상태(State)를 관측한 다음 작업 계획에 활용하기 위한 형식으로 변환한다. 세부 목표 예측 단계에서는 변환된 상태 정보를 바탕으로 초기 상태로부터 목표 상태에 도달하기 위해 로봇이 수행해야 할 세부 목표를 탐색한다. 정확한 작업 계획을 위해서는 환경 인식 단계에서 주어진 상태 정보를 유의미한 형태로 표현하는 과정이 중요하다. 이를 위해 다양한 방식의 연구가 활발히 이루어지고 있다.

과거에는 규칙 기반 방법을 활용한 작업 계획 연구가 주로 이루어졌다. 범용적으로 활용되는 Planning Domain Definition Language (PDDL)<sup>[3]</sup>은 표준화된 문법에 따라 작업 환경에 있는 물체의 종류나 행동의 실행 조건 및 결과를 도메인(Domain)에 정의하고, 물체를 설명하는 서술자(Predicates)의 집합으로 상태를 표현한다. 초기 상태를 목표 상태로 변화시키기 위해 필요한 세부 목표를 정의된 규칙에 따라서 탐색하기 때문에 기

Received : Mar. 8, 2024; Revised : Apr. 18, 2024; Accepted : Apr. 23, 2024

\* This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00907, Development of AI Bots Collaboration Platform and Self-organizing AI, 80% and No.RS-2020-II201373, Artificial Intelligence Graduate School Program (Hanyang University), 20%).

1. Master Student, Department of Electronic Engineering, Hanyang University, Seoul, Korea (bite3123@hanyang.ac.kr)

† Assistant Professor, Corresponding author: Department of Electronic Engineering, Hanyang University, Seoul, Korea (yoh21@hanyang.ac.kr)

준이 명확하고 이해하기 쉽다는 장점이 있다. 그러나 풀고자 하는 작업 조건에 맞는 규칙을 일일이 작성하는 과정이 필수적이기 때문에 다양한 환경 조건에서 범용적으로 사용하기 힘들다. 또한 물체가 많아질수록 탐색 공간이 기하급수적으로 커진다는 한계점이 존재한다.

인공지능 연구가 발전함에 따라 이를 작업 계획에 적용하는 연구가 활발히 이루어지고 있다. 특히 컴퓨터비전의 발전으로 주어진 환경 정보를 이미지로 관측하여 작업 계획의 입력(Input)으로 활용하는 연구가 진행되었다. [4]는 초기 상태와 목표 상태의 이미지를 입력으로 하여 수행해야 할 세부 목표를 출력하는 모델을 제안하였다. 이때 Convolution Neural Network (CNN) 기반의 모델을 설계하여 이미지 입력으로부터 유의미한 정보를 추출하였다. 이러한 방식은 여러 물체가 존재하는 상태 정보를 처리하는 과정이 간단하다는 장점이 있지만, 물체들 사이의 물리적 관계에 대한 정보는 활용되지 못하고 손실된다는 한계점이 있다. 최근에는 대량의 데이터로 사전 학습된 대규모언어모델(LLM)이 가지고 있는 추론 능력을 작업 계획에 활용하려는 시도가 활발히 이루어지고 있다. [5]는 술어(predicate)의 집합으로 표현된 초기 및 목표 상태 정보에 따른 작업 계획을 출력하도록 LLM을 프롬프팅하는 방법을 제안하였다. 그러나, 텍스트로 표현된 상태 정보만으로 물체 사이 관계를 이해해야 한다는 한계점이 존재한다.

이러한 문제를 해결하기 위해 최근에는 주어진 상태를 그래프로 표현하는 연구 또한 시도되고 있다. 이때 그래프 데이터의 특징 추출에 적합하다고 알려진 Graph Neural Network (GNN)<sup>[6]</sup>가 좋은 성능을 보이고 있다. GNN은 정점(Vertex)과 임의의 두 정점을 연결하는 간선(Edge)의 집합으로 이루어진 그래프 데이터의 구조적 특징을 고려하여 설계된 네트워크 구조이다. 각 정점에 입력된 특징(Feature)을 중심으로 주변에 연결된 이웃 정점과 간선의 특징들을 통합(Aggregation)하여 업데이트하는 메시지 패싱(Message Passing) 연산을 통해 유의미한 정보를 추출하도록 설계되었다. 이러한 장점으로 인해 작업 계획에 GNN을 활용하여 상태 정보를 추출하는 연구가 활발히 이루어지고 있다. [7]는 환경에 존재하는 물체들의 중요도를 예측하여 중요 물체에 대해서만 작업 계획을 하여 탐색 범위를 줄이는 방법을 제안하였다. 그러나 예측된 중요 물체들에 대해서 여전히 규칙 기반 작업을 수행하기 때문에 다양한 작업 환경에 적용하기 어렵다는 단점이 있다.

따라서 본 연구에서는 그래프로 표현된 상태 정보로부터 세부 목표를 예측하고, 그에 따른 상태 변화를 추정하면서 목표에 도달하기 위한 작업 계획을 수행하는 방법을 제안한다.

방법은 입력 상태에 대한 세부 목표 예측 모델과, 상태 변화 모듈을 탐색 알고리즘과 결합한 작업 계획 알고리즘으로 구성

된다. 먼저 작업 환경에 여러 개의 물체가 배치된 상태를 그래프로 표현한다. 현재 및 목표 상태의 그래프들을 하나의 그래프로 결합하고, 이를 세부 목표 예측 모델에 입력하여 수행할 가능성이 가장 높은 상위  $k$ 개의 세부 목표를 예측한다.  $k$ 개의 세부 목표에 따른 상태 그래프의 변화를 계산하고, 목표 그래프와의 유사도를 기준으로 탐색을 통해 작업 계획을 진행한다.

본 논문의 2장에서는 GNN 기반 세부 목표 예측 모델에 대해서 설명한다. 3장에서는 상태 변화 모듈과 최상 우선 탐색 알고리즘으로 구성된 작업 계획 시스템에 대해 설명한다. 4장에서는 제안된 방법의 검증에 위한 데이터셋 수집과 실험 결과를 설명한다. 5장에서는 본 논문의 결론 및 개선 방향에 대해 논의한다.

## 2. GNN 기반 세부 목표 예측 모델

2장에서는 주어진 현재 및 목표 상태를 하나의 결합된 그래프로 표현하고, 이를 입력으로 받아서 현재 상태에서 수행해야 할 세부 목표를 예측하는 GNN 기반 인공지능 모델을 제안한다.

### 2.1 상태 그래프 정의

상태 그래프는 주어진 상태에 존재하는 물체에 대한 정보와 물체 사이의 관계에 대한 정보를 포함한 유향 그래프(Directed Graph)로 정의된다. 그래프  $G = (\mathbf{V}, \mathbf{E})$ 에서 정점  $v \in \mathbf{V}$ 는 물체의 이미지에 대한 임베딩 벡터(embedding vector)로 표현한다. 본 논문의 실험에서 사용된 물체의 종류는 {'Table', 'Box', 'Gripper'} 3가지로 정한다. 각 물체의 이미지 임베딩은 전체 관측 이미지에서 해당 물체의 바운딩 박스('Bounding Box') 영역 이미지를 ResNet101<sup>[8]</sup> 인코더에 통과시켜서 구한다. 관측 이미지는 로봇의 hand-eye 시점으로 촬영하기 때문에, 그리퍼(gripper) 또는 그리퍼에 잡힌 물체의 이미지는 관측되지 않는다. 이러한 경우에는 이미지 임베딩을 영벡터(zero vector)로 대신 입력한다. 간선  $e \in \mathbf{E}$ 는 두 물체 사이의 의미적(semantic)인 관계 정보에 대한 원-핫 임베딩(one-hot embedding)으로 표현하며, 사용되는 관계의 종류는 {'On', 'Under', 'Grasp', 'In Grasp'}로 구성된다.

모델이 세부 목표를 예측하기 위해서는 현재 상태와 목표 상태에 대한 정보가 모두 필요하다. 이를 위해 본 연구에서는 두 상태를 표현한 각각의 그래프를 하나의 결합된 그래프로 표현하였다. 결합은 두 그래프에서 동일한 물체 쌍을 잇는 간선 벡터를 concatenation하는 방식으로 수행된다. 수식으로 표현하면 다음과 같다.

$$e_{concat} = e_{current} \oplus e_{goal} \quad (1)$$

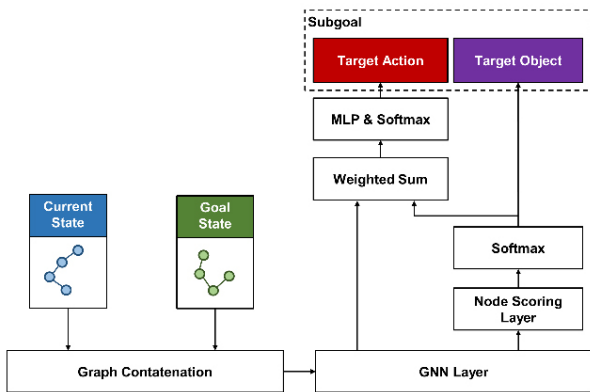
여기서  $e_{current}, e_{goal}$ 은 각각 현재 및 목표 상태에서 동일한 물체쌍을 연결하는 간선 벡터이며,  $\oplus$ 는 concatenation 연산을 의미한다. 어떤 두 물체를 잇는 간선이 한 시점에만 존재하는 경우에는 동일한 크기의 영벡터를 대신 concatenation 한다. 두 그래프에 존재하는 모든 간선에 대해 위 연산을 수행하여 그래프를 결합한다. 정점 벡터의 경우에는 해당 물체의 고유한 정보를 의미하기 때문에 현재와 목표 상태 사이에 변화가 유의미하지 않다. 따라서 물체 벡터는 결합하지 않고 현재 상태의 값으로 고정하여 사용한다. 이 과정을 통해 모델이 현재와 목표 상태 그래프 사이에 존재하는 연결성의 차이를 이해하여 로봇이 수행해야 할 세부 목표를 예측하도록 학습할 수 있다.

## 2.2 모델 구조

[Fig. 1]은 세부 목표 예측 모델의 구조를 표현한 그림이다. 먼저 입력된 상태 그래프를 *GNN layer*에 입력하여 그래프에 존재하는 모든 물체에 대한 정점의 임베딩 벡터를 업데이트한다. 해당 연산은 다음과 같이 식 (2)로 표현된다.

$$\begin{aligned} v' &= MP(v, N(v)) \\ &= MLP(v \oplus \sum (v_i^N, e_{v_i v}^N)) \end{aligned} \quad (2)$$

여기서  $MP$ 는 메세지 패싱 연산을 의미하며,  $N(v)$ 는 한 정점  $v$ 에 이웃한 정점의 집합이다.  $MLP$ 함수 내부의  $v_i^N$ 은  $N(v)$ 에 포함된 한 정점을,  $e_{v_i v}^N$ 은  $v_i^N$ 과  $v$ 를 잇는 지향성 간선을 의미한다. 그래프의 한 정점  $v$ 에 대해서 이웃한 정점의 정보와 그



[Fig. 1] Architecture of GNN-based subgoal prediction model. Current and goal state graphs are concatenated and then passed into the GNN layer to extract node (object) features. Finally, model predict the target action and object as subgoal

때의 연결 정보를 모두 합쳐서 정점  $v$ 와 concatenate하고, 이를  $MLP$ 함수에 통과시켜서 최종적으로 주변 정보를 포함하도록 업데이트된 정점 벡터  $v'$ 을 구할 수 있다. 이러한 연산을 통해 그래프에 존재하는 물체에 대해서 연결된 이웃 물체와 그와의 연결 정보를 모두 고려하여 유의미한 정보를 추출할 수 있다.

모델에 입력되는 상태에는 다수의 물체가 존재한다. 이 중에는 작업 계획 과정에서 주로 다루어지는 중요한 물체도 있지만, 목표에 따라서 어떠한 작업도 이뤄지지 않는 물체도 있다. 효율적인 작업 계획을 위해서는 목표 달성에 중요한 물체에 대해서만 세부 목표를 예측하여 탐색 공간을 줄이는 과정이 필수적이다. 이를 고려하기 위해서 본 연구에서는 업데이트된 그래프에 대해서 물체의 중요도를 예측하기 위해서 *Node scoring layer*를 추가하였다. 해당 레이어는 *GNN layer*와 구조와 연산은 동일하지만, 출력되는 물체 벡터의 차원을 1로 변경하고 *softmax*연산을 추가하여 모든 물체의 중요도에 대한 확률 분포를 예측한다. 이러한 과정을 통해 주어진 현재 및 목표 상태에서 세부 목표를 수행할 때 중요한 물체가 무엇인지 식별할 수 있다.

세부 목표 중 수행해야 할 행동(Action)의 종류를 예측하기 위해선 주어진 상태를 표현하는 하나의 통합된 벡터가 필요하다. 이를 위해 *GNN layer*를 거쳐서 업데이트된 물체 벡터들에 대해서 *node scoring layer*에서 예측된 중요도를 가중치로 한 가중합(Weighted sum)을 구한다. 이러한 과정을 통해 존재하는 모든 물체의 정보를 포함하면서, 세부 목표 예측에 중요한 물체를 더 많이 고려하는 유의미한 벡터를 구할 수 있다. 이후 *MLP layer*와 *softmax*를 추가하여 행동 공간(Action space)에서의 확률 분포를 예측한다. 예를 들어 Box 1, 2, 3이 테이블 위에 놓인 입력 상태와 Box 1이 2 위에 쌓인 목표 상태가 주어졌을 때, 목표 상태에 도달하려면 먼저 Box 1과 Table 사이에 연결된 'On'-'Under' 관계가 제거되고, Box 1과 Box 2 사이에 'On'-'Under' 관계가 추가되어야 한다. 주어진 목표를 만족하기 위해 로봇이 현재 상태에서 수행해야 할 행동은 Box 1을 집는 것이며, 따라서 이때 모델의 출력은 'Pick'과 'Box 1'이 된다. 이처럼 제안된 모델이 주어진 상황에서 중요 물체와 수행할 행동을 잘 예측하도록 학습하기 위해 랜덤하게 생성된 5000개의 작업 데이터셋을 수집하였다. 데이터 수집에 대한 자세한 내용은 4.1에서 설명한다. 본 논문에서는 로봇이 수행할 수 있는 행동의 종류를 {'Pick', 'Place'}의 두 가지로 정의하고 실험을 진행하였다.

모델의 학습을 위한 loss는 식 (3)과 같이 정의한다.

$$L_{Subgoal} = \omega_{Action} \cdot L_{Action} + \omega_{Object} \cdot L_{Object} \quad (3)$$

이때  $L_{Action}$  과  $L_{Object}$  는 정답 레이블(Ground truth label)에 대한 cross-entropy loss로 계산하였다. 각각의 loss에 대한 가중치는  $\omega_{Action} = 0.1$ ,  $\omega_{Object} = 1$ 로 설정하였다.

### 3. 작업 계획 시스템

3장에서는 주어진 사전 학습된 세부 목표 예측 모델을 바탕으로 목표에 도달하기 위한 연속적인 세부 목표의 목록을 계획하기 위한 방법을 설명한다.

#### 3.1 상태 변화 모듈

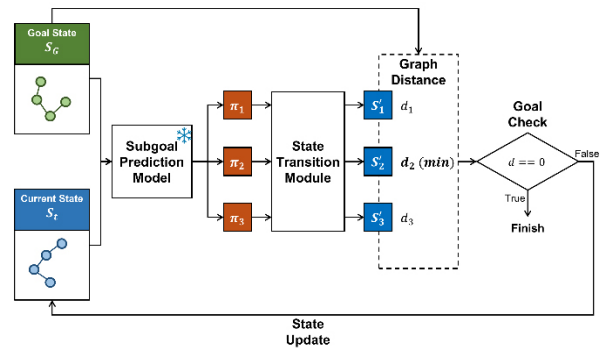
연속된 작업 계획을 위해서는 주어진 상태에서 특정 세부 목표를 수행했을 때 변화되는 다음 상태를 예측해야 한다. 상태 정보를 이미지 임베딩과 같은 암시적(Implicit)인 형태로 처리하면 상태 변화 예측을 위한 추가적인 모델 학습이 필요하다. 반면, 물체 사이의 관계를 그래프로 표현하는 경우에는 명시적(Explicit)인 관계성의 변화를 규칙적으로 정의할 수 있다. 따라서 본 논문에서는 과정의 간소화를 위해 입력된 현재 상태의 그래프에서 특정 행동에 따라 대상 물체의 연결성 변화를 계산하는 규칙을 정의하였다.

예를 들어 Box1, 2, 3이 테이블 위에 놓인 현재 상태에서 세부 목표로 ('Pick', 'Box 1')이 주어진 경우, 그래프에서 Box 1과 연결된 모든 간선을 지우고 그리퍼와 'Grasp'-'In Grasp' 간선을 추가한다. 또한 Box 1이 잡힌 상태에서 ('Place', 'Box 2')라는 세부 목표가 주어지면 그리퍼와 연결된 간선을 지우고 'Box 1'과 'Box 2' 사이에 'On'-'Under' 간선을 추가한다. 이러한 규칙으로 현재 상태 그래프에 대해 임의의 세부 목표가 주어진 경우, 그때 변화할 다음 상태 그래프를 실제 로봇 수행 없이도 예측할 수 있다.

#### 3.2 최상 우선 탐색 알고리즘

작업 계획 성능의 향상을 위해서는 여러 경우를 고려하여 세부 목표를 탐색해야 한다. 이때 임의의 행동을 선택하면서 탐색하는 것은 탐색 공간이 너무 커지기 때문에 비효율적이다. 따라서 본 연구에서는 [Fig. 2]와 같이 학습된 모델과 탐색 알고리즘을 결합한 작업 계획 방법을 제안한다. 먼저 앞서 사전 학습된 세부 목표 예측 모델을 통해 특정 시점에서 수행해야 하는 상위 k개의 세부 목표에 대해서 탐색을 진행한다.

상태 변화 모듈을 거쳐서 각각의 세부 목표에 대해 변화된 다음 상태를 예측하고, 그중 다음 탐색을 진행할 상태를 선정하여 위 과정을 반복한다. 다음 탐색 상태의 선정 기준인  $d_{graph}$



[Fig. 2] Search algorithm with pretrained subgoal prediction model

(graph distance)는 다음과 같은 식 (4)으로 구해진다.

$$d_{graph} = n(E_{Goal} - E_{Current}) \tag{4}$$

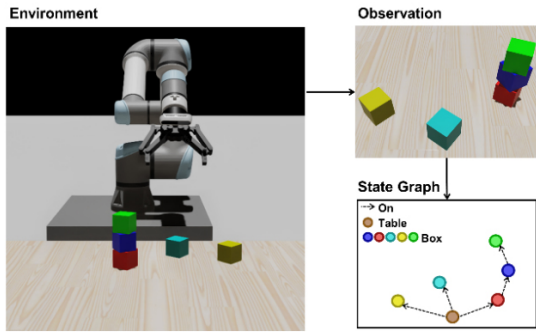
여기서  $E_{Goal}$ ,  $E_{Current}$  는 각각 목표 그래프와 현재 그래프에 포함된 간선의 집합이다. 두 집합의 차집합에 포함된 요소 개수를 통해 현재 상태의 그래프와 목표 그래프 사이의 차이를 산술적으로 표현할 수 있다. 이를 바탕으로 탐색 트리의 모든 단말 노드(Leaf node) 중에서  $d_{graph}$  가 최소인 노드를 다음 탐색 노드로 선정한다. 위 과정을 반복하면서  $d_{graph}$  가 0인 노드가 나타난 경우, 해당 노드를 목표 상태에 도달한 노드로 처리하고, 초기 상태 노드로 탐색 경로를 역추적해서 수행된 세부 목표의 목록을 구한다. 이러한 탐색을 통해 작업 계획의 성능을 향상시킬 수 있음을 4장에서 설명한다.

## 4. 실험 및 결과

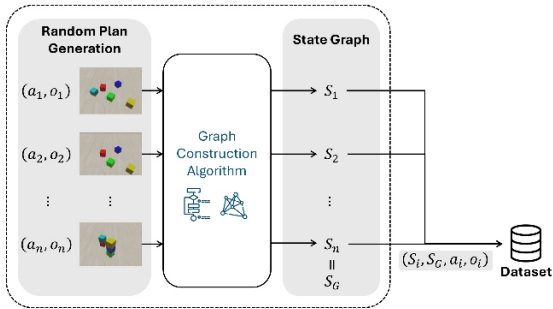
#### 4.1 실험 환경

세부 목표 예측 모델의 학습을 위한 데이터셋을 수집하기 위해 Isaac sim 시뮬레이터로 [Fig. 3]과 같이 환경을 구축하였다. 이렇게 구축된 환경에서 작업 데이터를 생성하기 위해 [Fig. 4]와 같은 작업 데이터 수집 시스템을 구성하였다. 먼저 5개의 box들을 랜덤하게 배치하여 초기 상태를 지정하고, 랜덤하게 선택된 물체  $o_i$ 에 대해 동작  $a_i$ 을  $n$ 번 수행하여 최종적으로 물체가 배치된 상태를 목표 상태  $S_G$ 로 수집하였다. 이때 각 시점에서의 물체 배치에 따른 상태 그래프  $S_t$ 를 구하기 위해서 물체들의 3차원 위치 좌표  $(x, y, z)$ 를 바탕으로 그래프를 구성하는 알고리즘을 구현하였다. 어떤 두 물체의 2차원 좌표  $(x, y)$ 가 오차범위 내에 겹치는 경우,  $z$ 좌표에 따라 'On', 'Under' 간선을 추가하였다. 또한 그리퍼가 어떤 물체를 집어





[Fig. 3] Simulation environment setting



[Fig. 4] Random stacking task data collection in simulation

서 관측 데이터에 보이지 않는 경우에는 ‘Grasp’, ‘In Grasp’ 간 선을 추가하고, 해당 물체의 정점 벡터는 영벡터로 대체하였다. 이때 각 물체들의 위치 좌표는 시뮬레이터로부터 직접 수집하였다. 생성된 작업 데이터에서 특정 시점의 상태  $S_i$ 와 그 때 수행된 세부 목표  $(a_i, o_i)$ , 그리고 최종 상태  $S_G$ 를 하나의 단위로 묶어서 세부 목표 예측 모델을 학습하였다. 작업의 길이  $n$ 은 2-8 범위에서 랜덤하게 지정하여 다양한 길이의 작업 데이터를 수집하였다. 최종적으로 5,000개의 작업 데이터에 대해서 22,392개의 학습 데이터를 수집하였다. 학습 및 검증을 위해 train, test (validation)의 비율을 7:3으로 분할하여 사용하였다. 평가를 위한 세부 지표는 다음과 같이 정의하였다.

- Plan Success Rate (PSR): 작업 계획 모델의 예측 계획 한 성공률(로봇 실행은 미포함).
- Execution Success Rate (ESR): 모델에서 예측한 계획으로 로봇이 수행한 결과에 대한 성공률.
- Additional Search Ratio (ASR): 모델에서 예측된 계획의 길이와 정답 계획의 길이의 비율로, 모델이 예측한 계획에 불필요한 과정이 얼마나 포함되었는가를 의미함.

#### 4.2 작업 계획 성능 실험

앞서 설명한 세부 목표 예측 모델과 상태 변화 모듈을 활용

한 탐색방법의 성능을 검증하기 위해 2가지 비교 모델과 비교 실험을 진행하였다. 비교 모델에 대한 설명은 다음과 같다.

- GNN: 세부 목표 예측 모델 출력 중 가장 확률이 높은 결과에 대해서만 상태를 업데이트하면서 목표에 도달할 때까지 반복.
- LLM<sup>[5]</sup>: 작업 도메인 정보와 초기 및 목표 상태 정보를 입력 받아서 수행해야 할 세부 목표의 목록을 출력하도록 LLM모델을 프롬프팅하여 작업을 계획함.
- Random Search: 랜덤하게 선택한 k개의 세부 목표에 대해 탐색.
- GNN Search (Ours): 세부 목표 예측 모델 출력 중 확률이 높은 상위 k개의 세부 목표에 대해 탐색.

2장에서 제안한 세부 목표 예측 모델의 유무에 따른 성능 차이를 확인하기 위해 Random Search 모델을 선정하였다. [Fig. 2]에서 가능성이 높은 상위 k개의 세부 목표를 예측하는 모델 부분을 대신하여 수행 가능한 행동과 물체를 랜덤하게 k개 선정하는 방식으로 작업을 계획하였다. 또한 3장에서 제안한 탐색 알고리즘 유무에 따른 성능을 평가하기 위해 GNN 모델과의 비교를 진행하였다. 이를 위해 세부 목표 예측 모델과 상태 변화 모듈을 가지고 탐색 없이 가장 높은 확률의 세부 목표에 대해서 그래프를 업데이트 하는 방식으로 작업을 계획하였다. 그리고 본 논문에서 제안한 방법이 물체 사이의 물리적인 관계 정보를 잘 이해할 수 있음을 보이기 위해 최근 작업 계획 분야에서 많이 연구되는 LLM 기반 방법<sup>[5]</sup>과 성능을 비교하였다. 실험을 위해 작업 환경에 있는 물체 및 동작의 종류와 같은 도메인 정보와 논리 술어의 집합으로 표현된 초기 및 목표 상태 정보를 LLM모델에 입력해서 필요한 세부 목표의 목록을 출력하도록 프롬프트를 작성하였다. 이때 논문에서 제시한대로 2개의 예시를 제공하는 few-shot 프롬프팅 기법을 적용하였고, LLM모델은 GPT-4를 사용하였다. 본 논문에서 제안하는 방법과 비교 모델 3가지를 포함한 총 4가지 모델에 대해 작업 계획 성공률(Plan Success rate)을 측정하였다. Random Search와 GNN Search에 대해서는 탐색할 세부 목표의 수 k를 3으로, 탐색 깊이(Search depth)제한을 10으로, 총 탐색 횟수를 20회로 제한하였다.

[Table 1]은 비교 모델에 대한 작업 계획 성공률 비교표이다. 실험 결과, 본 논문에서 제시한 GNN Search가 가장 높은 성공률을 기록하였다. 특히 GNN 모델과 비교했을 때 큰 폭으로 성능 차이가 존재함을 확인할 수 있다. 이는 여러 세부 목표를 고려하지 않으면 틀렸을 경우 다른 경로로 재계획할 수 없기 때문이다. LLM기반 방법에 대해서도 약 41% 높은 성공률을 보였는데, 주어진 환경 정보를 모두 텍스트로 표현해야 하

[Table 1] Model prediction results on test dataset

	GNN	LLM	Random Search	GNN Search
PSR	0.294	0.538	0.877	<b>0.949</b>

[Table 2] Simulation experiment results

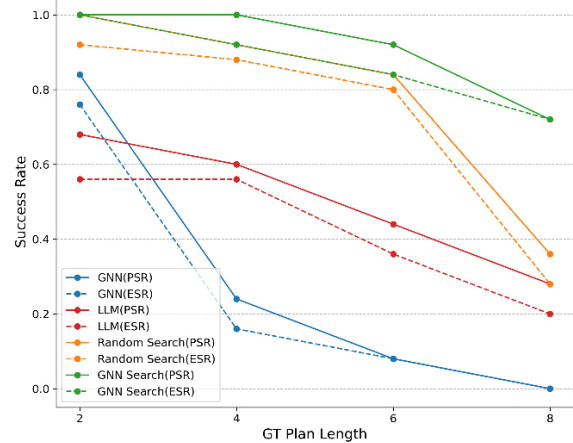
	GNN	LLM	Random Search	GNN Search
PSR	0.29	0.50	0.78	<b>0.91</b>
ESR	0.25	0.42	0.72	<b>0.87</b>
ASR	<b>1.000</b>	1.093	1.548	1.104

는 LLM과 다르게 본 논문의 방법은 그래프 구조로부터 물체 사이의 관계 정보를 잘 이해할 수 있기 때문에 추측된다. Random Search와 비교했을 때도 약 7% 정도 높은 성능을 기록하였다. 이는 사전 학습된 세부 목표 예측 모델이 목표에 도달할 가능성이 높은 방향으로 탐색을 유도하기 때문이다. 따라서 본 논문에서 제시한 방법이 재계획이 가능하면서 탐색의 효율도 높았음을 알 수 있다.

### 4.3 시뮬레이션 실험

제안된 방법의 실제 로봇 적용 가능성을 검증하기 위해 시뮬레이션 환경에서의 실험을 진행하였다. 시뮬레이션 환경은 NVIDIA GeForce RTX 3060 GPU가 탑재된 워크스테이션에 구축하였다. 시뮬레이터는 Isaac Sim 2023.1.1 버전을 사용하였고, 로봇은 Universal Robots의 UR5e를 사용하였다. 또한 그리퍼는 Robotiq사의 2F-140을 사용하였고, 환경 관측을 위한 Intel Realsense D-435 카메라 센서를 그리퍼 위에 부착하였다. 실험 환경은 데이터 수집 과정과 동일하게 서로 다른 색상의 박스 5개와 작업 테이블로 구성하였고, 실험마다 박스의 배치와 목표를 랜덤하게 설정하였다. 작업 계획 성능에 대해서만 평가하기 위해 물체의 위치 정보는 알고 있다고 가정하였다. 또한 로봇의 모션 제어는 오픈소스 라이브러리인 Moveit!에서 제공하는 RRT기반<sup>9)</sup>의 모션 계획기를 활용하였다.

[Table 2]는 비교모델별 시뮬레이션 실험 성공률 비교 결과이다. 여기서 실험 결과 앞서 4.2장과 마찬가지로 본 논문에서 제시한 방법이 모든 지표에서 가장 좋은 성능을 기록하였다. 전반적으로 ESR이 PSR에 비해 낮은 수치를 보였는데, 이는 모델에서 예측된 계획을 로봇이 수행하는 과정에서 모션 생성의 부정확함 때문으로 추정된다. 그러나 PSR에 대한 ESR의 성능 하락 폭은 Random Search가 Ours보다 더 크게 나타났다. 이는 모델에서 불필요한 세부 목표를 계획에 더 많이 포함할수록 모션 생성 실패 가능성이 함께 올라가기 때문이다. ASR



[Fig. 5] Simulation experiment results for task step length

만 비교했을 때는 GNN이 1.0으로 가장 높은 결과를 보였지만, PSR과 ESR은 각각 0.29와 0.25로 매우 낮은 수치를 기록하였다. 이는 탐색 알고리즘 없이는 작업 목표까지의 길이가 짧거나, 세부 목표 예측 모델이 학습했던 작업과 유사한 경우에만 문제를 풀 수 있었기 때문으로 추정된다. LLM의 경우에는 Ours와 비슷한 ASR을 기록했지만 PSR과 ESR에 대해서는 대략 40%정도 낮은 수치를 보였다. 이는 LLM모델이 few-shot 프롬프팅으로 제공되는 예시 작업을 바탕으로 작업을 계획하기 때문에 예시에 비해 너무 긴 계획은 생성되지 않기 때문으로 보인다. 그러나 물체 사이 물리적 관계 정보를 정확히 이해하지 못하는 결과가 생성되는 경우가 있었다. 예를 들어 Box 1이 Box 2 위에 놓인 경우에는 Box 2를 집을 수 없다는 조건이 주어졌음에도 불구하고 Box 2를 Pick하거나, 어떤 Box를 집고 있는 상태에서 다른 Box를 집는 계획이 만들어졌다. 반면 제안된 방법은 그래프 구조의 상태로부터 관계 정보를 잘 이해할 수 있고, 작업 계획 알고리즘에서 목표 상태로 도달하는 방향으로 세부 목표를 선택하기 때문에 매 순간 적절한 동작을 계획할 수 있다. 따라서 본 논문의 방법이 탐색을 통해 작업 계획 성공률을 높이면서, 사전 학습된 세부 목표 예측 모델을 통해 탐색을 효율적으로 수행하여 불필요한 계획을 줄일 수 있음을 확인하였다.

다음으로 [Fig. 5]에서 GT Plan Length에 따른 결과를에서 비교하였다. 여기서 GT Plan Length는 정답 계획의 길이를 의미한다. 전반적으로 정답 계획의 길이가 증가할수록 PSR과 ESR이 감소하는 모습을 보였다. GNN모델의 경우에는 대부분 길이가 2 또는 4로 짧은 작업에 대해서만 성공하였음을 확인할 수 있다. 이는 모델이 학습한 데이터와 유사한 작업에 대해서만 잘 계획할 수 있기 때문으로 추정된다. 학습 모델 단독으로는 세부 목표 예측의 최적성이 보장되지 않기 때문에 탐색 알고리즘을 통한 보완이 필요하다. 정답 계획 길이의 증가에 따른 PSR 감소 정도를 비교한 결과, 다른 비교모델들에 비해

Ours가 가장 적게 감소하였다. 따라서 본 논문의 방법이 긴 과정이 필요한 작업에 대해서도 성능을 잘 유지할 수 있음을 보였다.

## 5. 결론

본 논문에서는 로봇의 작업 계획에 필요한 상태 정보를 그래프로 표현하여 물체 사이의 관계에 대한 정보를 명시적으로 활용하는 방법을 제안하였다. 또한 주어진 현재 상태와 목표 상태의 그래프를 결합하여 모델이 두 상태 사이의 차이점을 학습하도록 만들었다. 세부 계획을 예측할 때 필요한 상태 정보를 그래프로부터 추출하기 위해 GNN 기반의 모델을 새롭게 제안하였다. 주어진 작업 목표에 대해서 현재 존재하는 물체들의 중요도를 예측하고, 이를 고려하여 세부 목표를 구성하는 행동 및 대상 물체를 예측함으로써 작업 계획의 효율을 높였다. 또한 작업 계획의 안정성을 높이기 위해 여러 가능성을 고려하는 탐색 방법을 결합하였다. 실험을 통해 본 논문에서 제시한 방법이 모델의 예측이 잘못된 경우에 재계획을 통해 목표를 달성할 수 있음을 보였다. 또한 사전 학습된 세부 목표 예측 모델을 결합하여 탐색을 효율적으로 수행하였다. 최종적으로 가상 환경에서 로봇 실험을 통해 제안된 방법이 실제 활용할 수 있음을 확인하였다. 추후 상태 변화 모듈을 학습 기반으로 구성하여 더 다양한 경우에 적용 가능하도록 할 계획이다. 또한 물체의 수와 종류를 다양하게 구성하여 모델의 환경에 대한 일반화 능력을 검증할 예정이다.

## References

[1] M. D. Graaf, S. B. Allouch, and J. V. Diik, "Why do they refuse to use my robot?: Reasons for non-use derived from a long-term home study," *2017 ACM/IEEE International Conference on Human-Robot Interaction*, Vienna, Austria, pp. 224-233, 2017, DOI: 10.1145/2909824.3020236.

[2] H. Guo, F. Wu, Y. Qin, R. Li, K. Li, and K. Li, "Recent Trends in Task and Motion Planning for Robotics: A Survey," *ACM Comput. Surv.*, vol. 55, no. 13, pp. 1-36, Jul., 2023, DOI: 10.1145/3583136.

[3] M. Ghallab, C. A. Knoblock, D. E. Wilkins, A. Barrett, D. Christianson, M. T. Friedman, C. Kwok, K. Golden, S. Penberthy, D. E. Smith, Y. Sun, and D. Weld, "PDDL - The Planning Domain Definition Language," *Technical report, Yale Center for Computational Vision and Control*, 1998, [Online], <https://www.researchgate.net/publication/2278933>.

[4] C. Paxton, Y. Barnoy, K. Katyal, R. Arora, and G. D. Hager, "Visual robot task planning," *2019 International conference on robotics and automation (ICRA)*, pp. 8832-8838, 2019, DOI:

10.1109/ICRA.2019.8793736.

[5] T. Silver, V. Hariprasad, R. S. Shuttleworth, N. Kumar, T. Lozano-Perez, and L. P. Kaelbling, "PDDL planning with pre-trained large language models," *NeurIPS 2022 foundation models for decision making workshop*, 2022, [Online], <https://openreview.net/forum?id=1QMMUB4zfl>, Accessed: Apr. 23, 2024.

[6] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI open*, vol. 1, pp.57-81, 2020, DOI: 10.1016/j.aiopen.2021.01.001.

[7] T. Silver, R. Chitnis, A. Curtis, J. Tenenbaum, T. Lozano-Perez, and L. P. Kaelbling, "Planning with learned object importance in large problem instances using graph neural networks," *Proceedings of the AAAI conference on artificial intelligence*, pp. 11962-11971, Sept., 2021, DOI: 10.48550/arXiv.2009.05613.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778, 2016, DOI: 10.48550/arXiv.1512.03385.

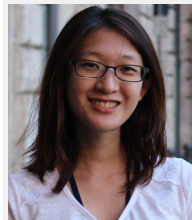
[9] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, San Francisco, CA, USA, pp. 995-1001, 2000, vol. 2, DOI: 10.1109/ROBOT.2000.844730.



### 변성완

2022 성균관대학교 기계공학부(학사)  
2022~현재 한양대학교 융합전자공학부  
(석사과정)

관심분야: Task Planning, AI



### 오윤선

2011 서울대학교 전기정보공학부(학사)  
2018 서울대학교 전기컴퓨터공학부(박사)  
2018~2019 Brown University 박사 후 연구원  
2019~2021 한국과학기술연구원  
지능로봇연구단 연구원

2021 한국과학기술연구원 지능로봇연구단 선임연구원  
2021~현재 한양대학교 융합전자공학부 조교수

관심분야: Task Planning, Human Robot Interaction