OSAKA UNIVERSITY

HARADA LAB
Robotic Manipulation

Efficient Task/Motion Planning for a Dual-arm Robot From Language Instruction and Cooking Images

---

Kota Takata*, Takuya Kiyokawa*, Ixchel G. Ramirez-Alpizar**
Natsuki Yamanobe**, Weiwei Wan* and **Kensuke Harada***,**

*Osaka University
** National Inst. Advanced Industrial Science and Technology

Introduction

ARADA LAB
Robotic Manipulation

Lack of Labor in Food Industry
⇒ Automation is expected

[1]https://newswitch.jp/p/5334

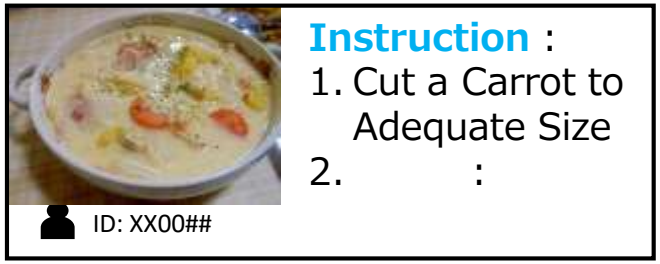
Manual Motion Teaching

↓

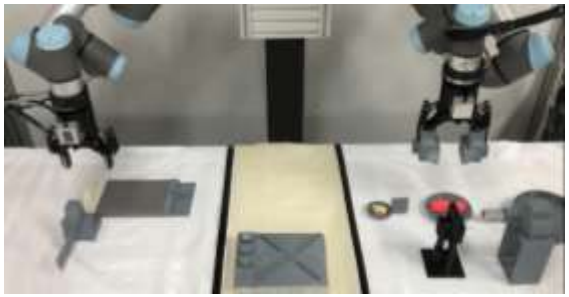
Automatic Motion Generation

Dual-arm Manipulation Planning from Cooking Recipe with Information Completion

Overview of Our Approach

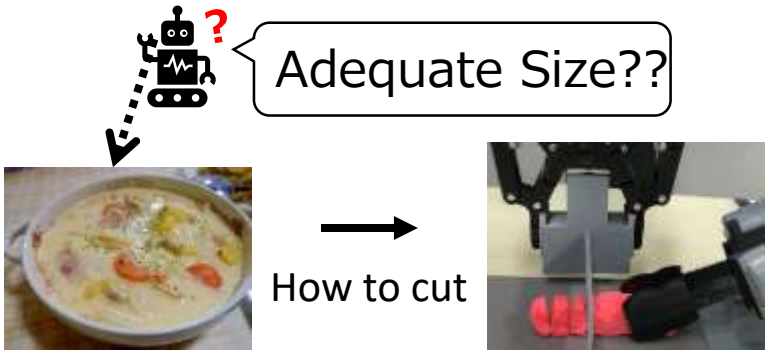


Cooking Recipe
(Picture + Verbal Instruction)

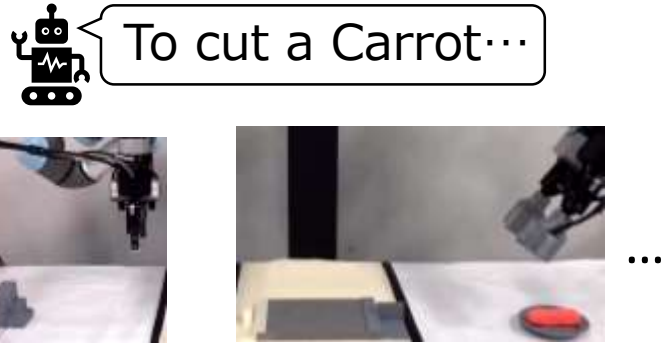


Cooking Motion Planning

Information Completion



Completion from Food Image

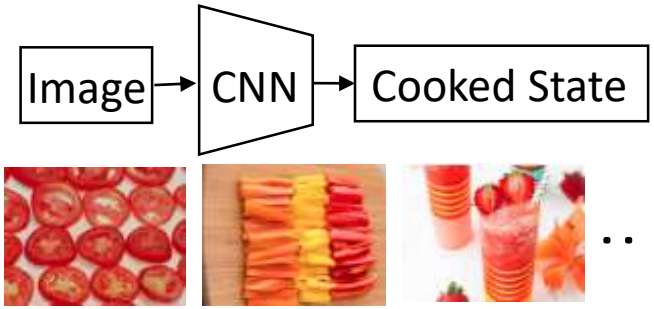


Completion of Preparation Motion

Related Works

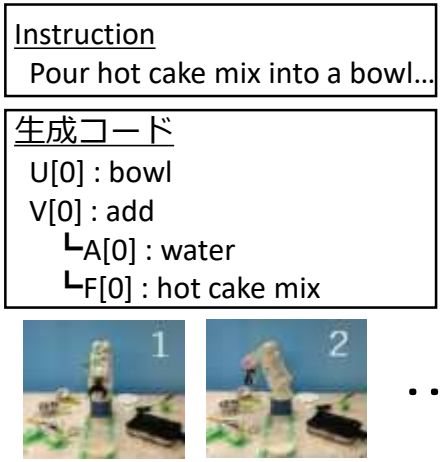
ARADA LAB
Robotic Manipulation

- Estimation of Food State
  - Rahul Paul. (2018)
  - MS Salekin et al. (2019)



No research on Cooked State Estimation

- Cooking Task Planning from Recipe
  - Kunze et al.(2015)
  - Inagawa et al. (2021)
    - ➢ Rule based approach
    - ➢ Single arm without motion planning



This Research

- Consideration of motion which is not explicitly written in a recipe.
- Efficient graph-based motion planning with a dual-arm manipulator

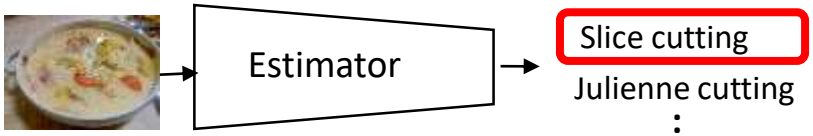
Method Overview

ARADA LAB
Robotic Manipulation

1st Step

Information Completion from Food Image

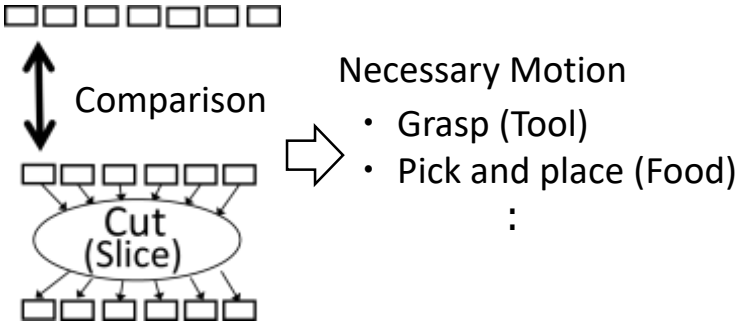
Estimation of Food State



2nd Step

Information Completion of What is not Explicitly Written in Recipe

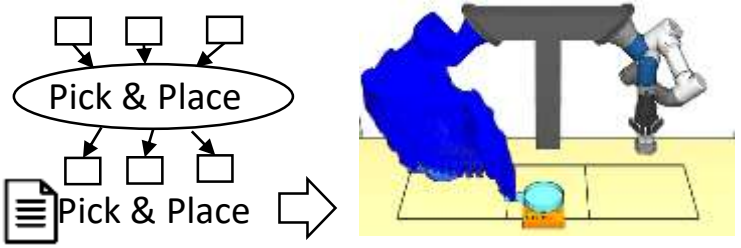
Comparison of Input/Output of Graph Elements



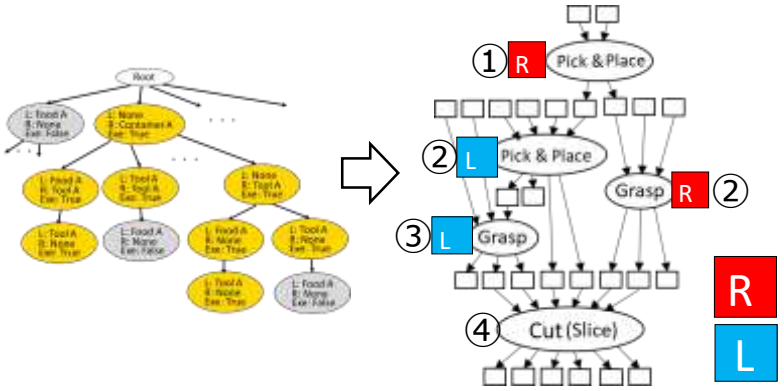
3rd Step

Motion Planning

by using Information obtain from the Graph (e.g., pick and place)



Connection of Graph Elements for Efficient Bimanual Task Planning



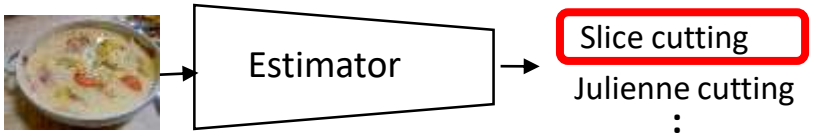
Method Overview

ARADA LAB
Robotic Manipulation

1st Step

Information Completion from Food Image

Estimation of Food State



2nd Step

Information Completion of What is not Explicitly Written in Recipe

Comparison of Input/Output of Graph Elements

3rd Step

Motion Planning

by using Information obtain from the Graph (e.g., pick and place)

Connection of Graph Elements for Efficient Bimanual Task Planning

Method Overview

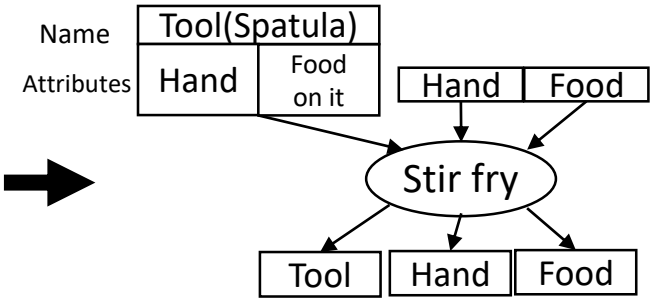
Recipe

1. Saute pork with high heat on a frypan
2. :

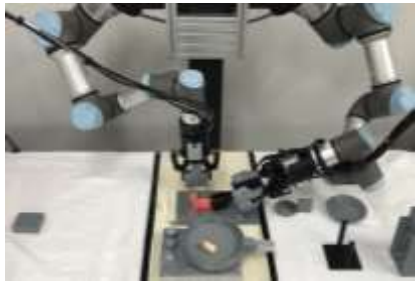
Motion Frame[2][3]

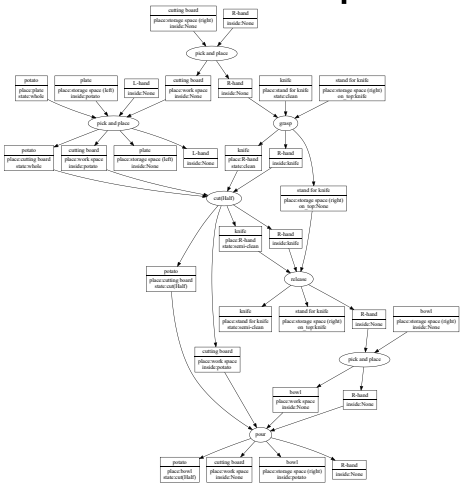
Stir fry	
Food	Pork
Type	High
Tool	Spatula
Container	Cooking pan
Equipment	Stove

Motion Graph Element



Motion Graph





[2] Das, Dipanjan, et al. “Frame-semantic parsing.” (2014):
[3] Haonan Chen, et al. "Enabling Robots to Understand Incomplete Natural Language Instructions Using Commonsense Reasoning." (2019).

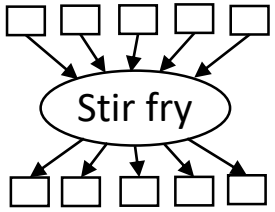
ARADA LAB
Robotic Manipulation

Lacking Information

■ Enough Information Available

1. Saute pork with high heat on a frypan
2. :

Stir fry	
Food	Pork
Type	High
Tool	Spatula
Container	Cooking pan
Equipment	Stove

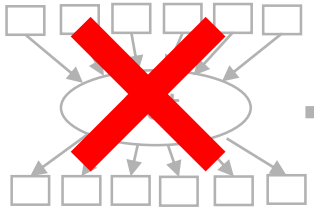


Robot Motion

■ Enough Information NOT Available

1. Cut carrot into adequate size.
2. :

Cut	
Food	Carrot
Type	???
Tool 1	Knife
Tool 2	Tong
Container	Cutting board



Robot Motion

Lacking Information
➡ Graph structure used for motion planning cannot be generated

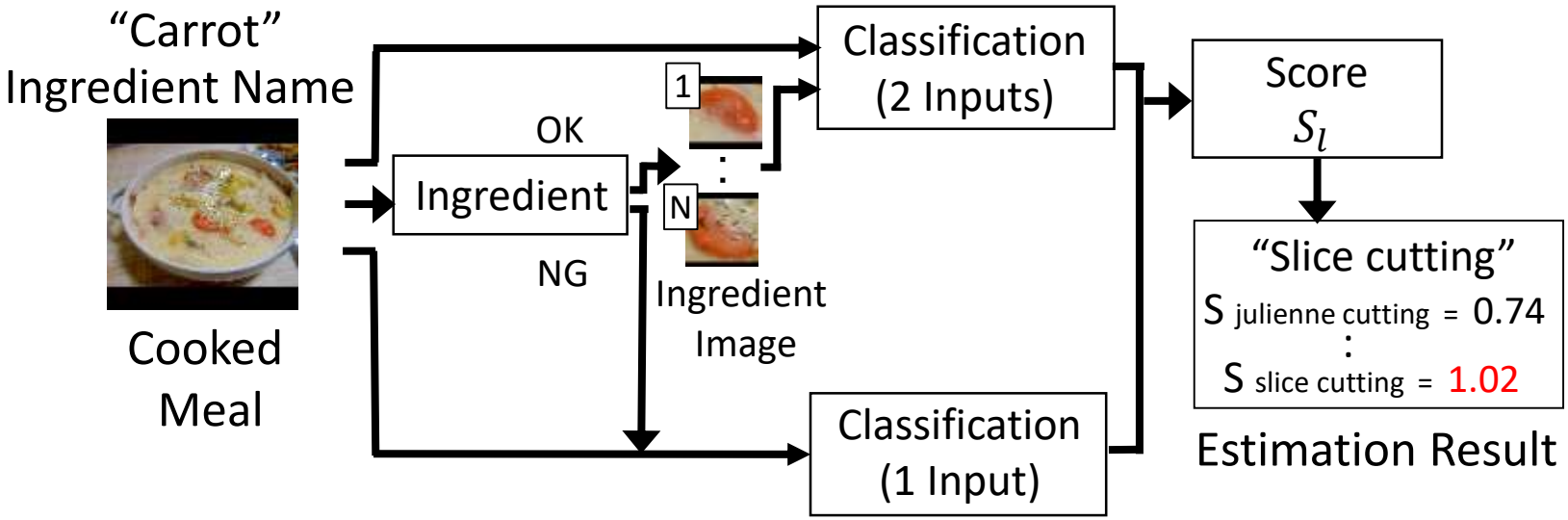
[2] Das, Dipanjan, et al. “Frame-semantic parsing.” (2014):
[3] Haonan Chen, et al. "Enabling Robots to Understand Incomplete Natural Language Instructions Using Commonsense Reasoning." (2019).

ARADA LAB
Robotic Manipulation

Estimation of Cooking State from Food Image

※ Lacking Information on Cutting Method
※Estimation of Cutting Method

- Two Stage Estimator : Ingredient ⇒ Cutting State
- Image of Cooked Meal + Included Ingredient



Method Overview

ARADA LAB
Robotic Manipulation

1st Step

Information Completion from Food Image

Estimation of Food State

3rd Step

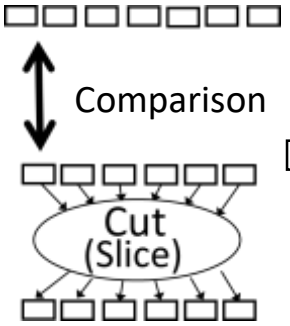
Motion Planning

by using Information obtain from the Graph (e.g., pick and place)

2nd Step

Information Completion of What is not Explicitly Written in Recipe

Comparison of Input/Output of Graph Elements

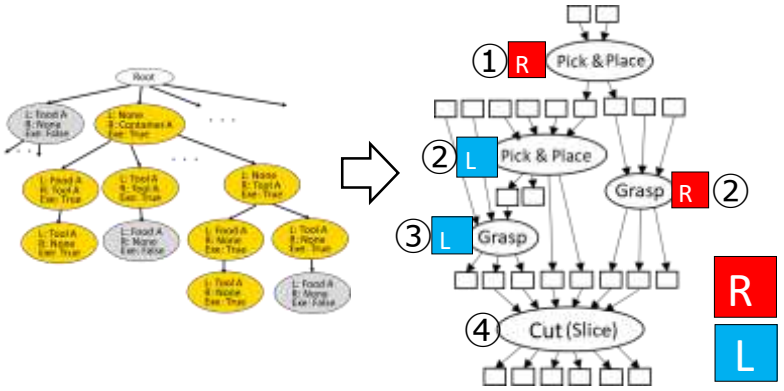


Necessary Motion

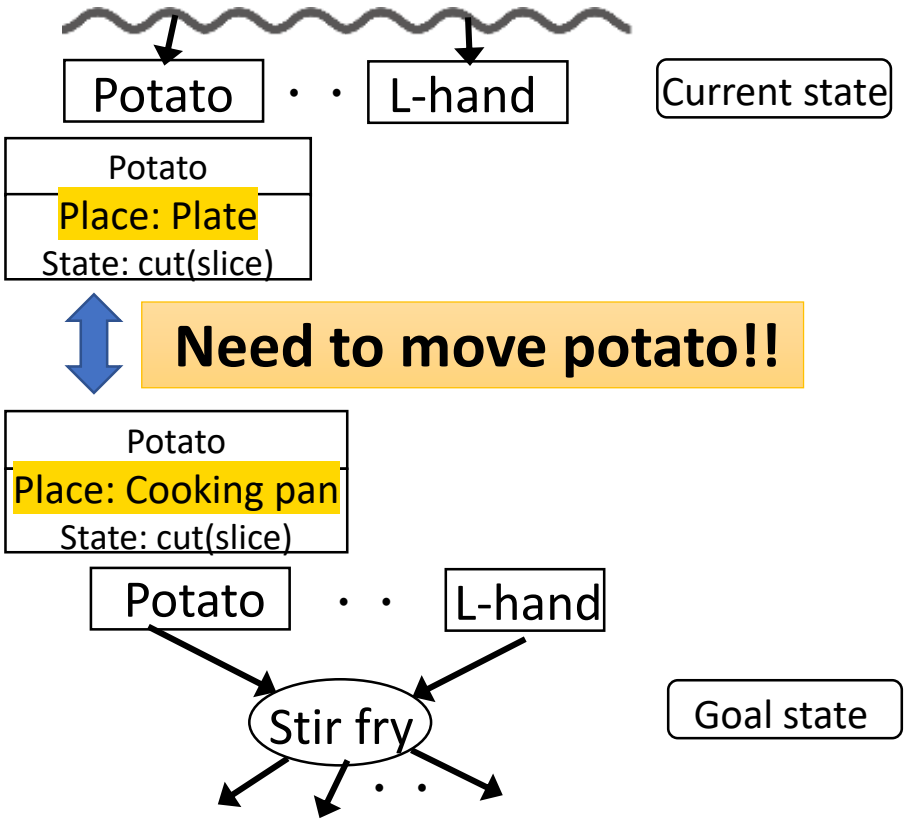
- Grasp (Tool)
- Pick and place (Food)

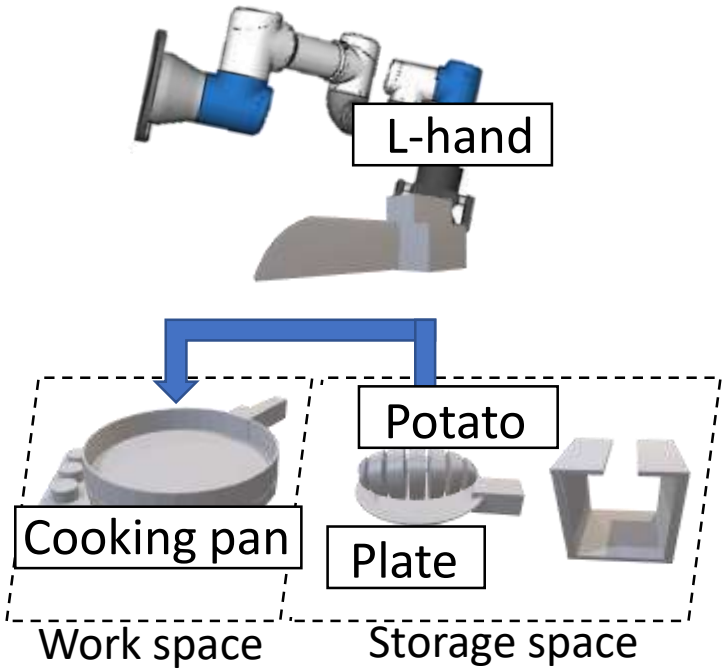
:

Connection of Graph Elements for Efficient Bimanual Task Planning



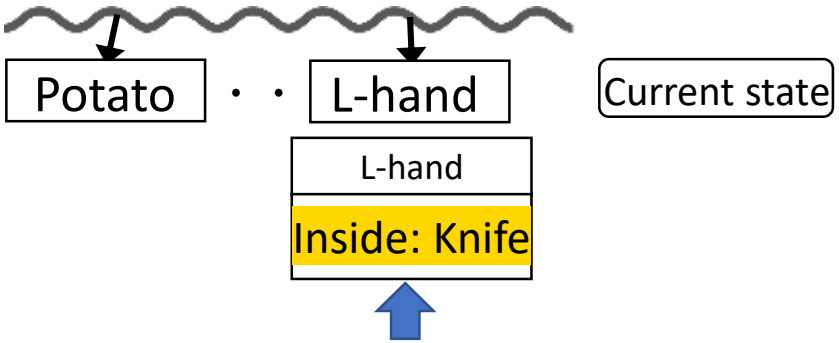
Determine Necessary Motion from Node Attribution[4][5]



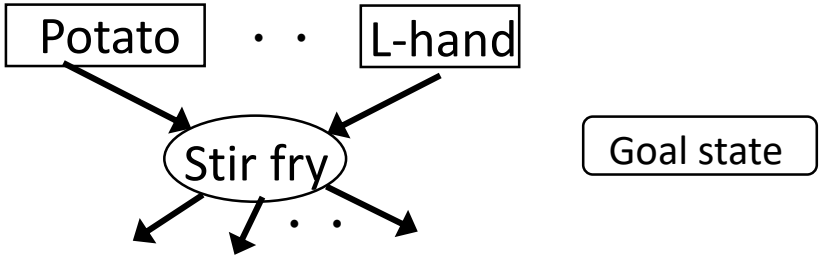


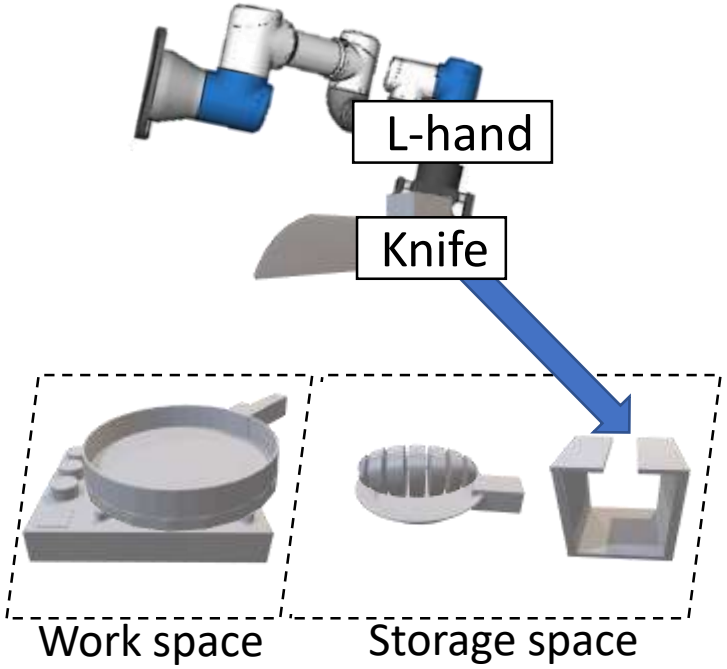
[4] PAULIUS et al. “Functional object-oriented network for manipulation learning,” (2016)
[5] PAULIUS et al. “A Weighted Functional Object-Oriented Network for Task Planning,” (2019)

Determine Necessary Motion from Node Attribution[4][5]



▶Knife should be brought back to shelf



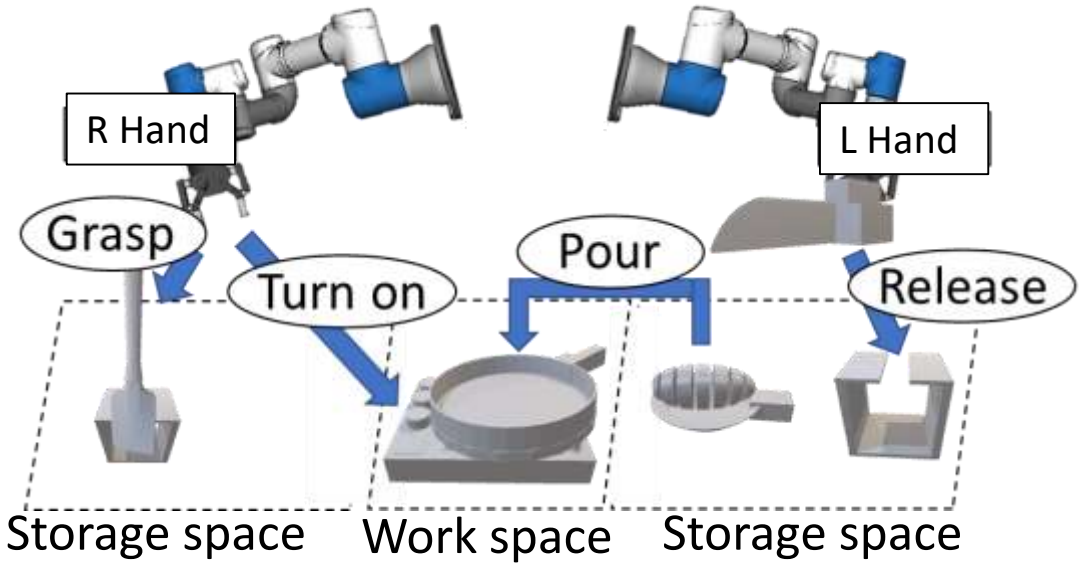


[4] PAULIUS et al. “Functional object-oriented network for manipulation learning,” (2016)
[5] PAULIUS et al. “A Weighted Functional Object-Oriented Network for Task Planning,” (2019)

ARADA LAB
Robotic Manipulation

List of Necessary Motion

Task R: Task Performed by R Arm
Task L: Task Performed by L Arm
Task RL: Task Performed by either R or L



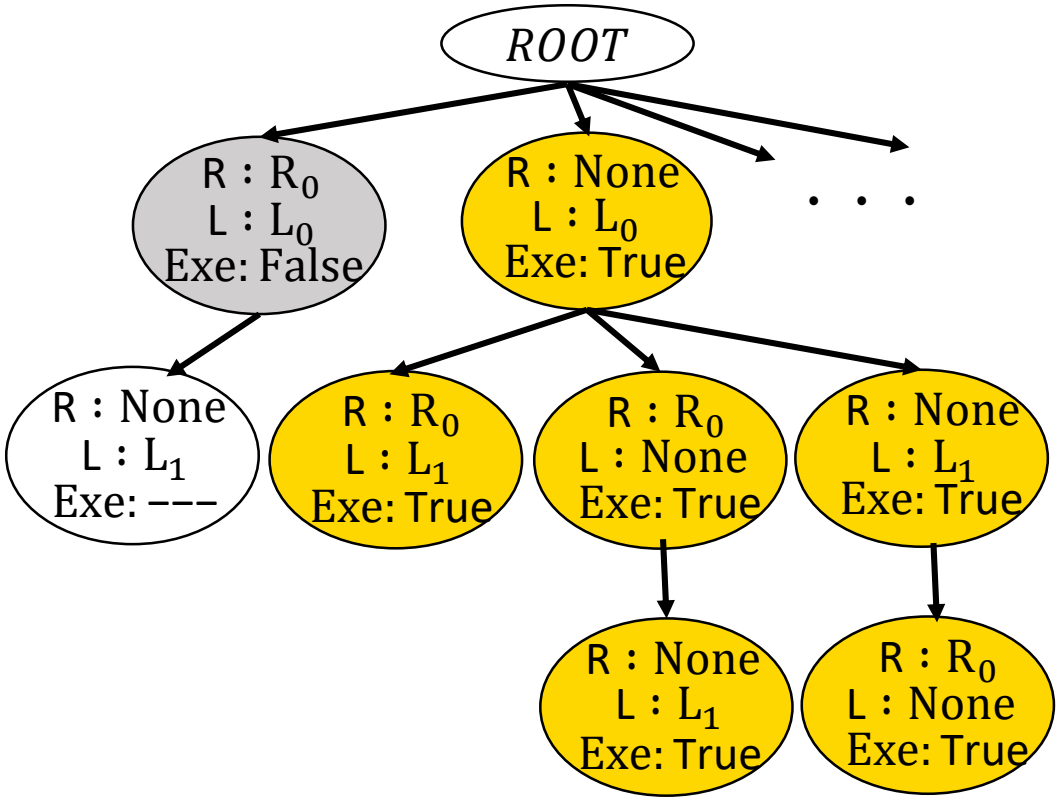
Task R =
{Turn on(stove), Grasp(tool)}
Task L =
{Release(tool), Pour(food)}
Task RL = { }

ARADA LAB
Robotic Manipulation

Sequence Graph

Tree Structure for Task Allocation

例. Task R = $\{R_0\}$
Task L = $\{L_0, L_1\}$



Node

- R：Task of R Arm
- L：Task of L Arm
- Exe：Executable

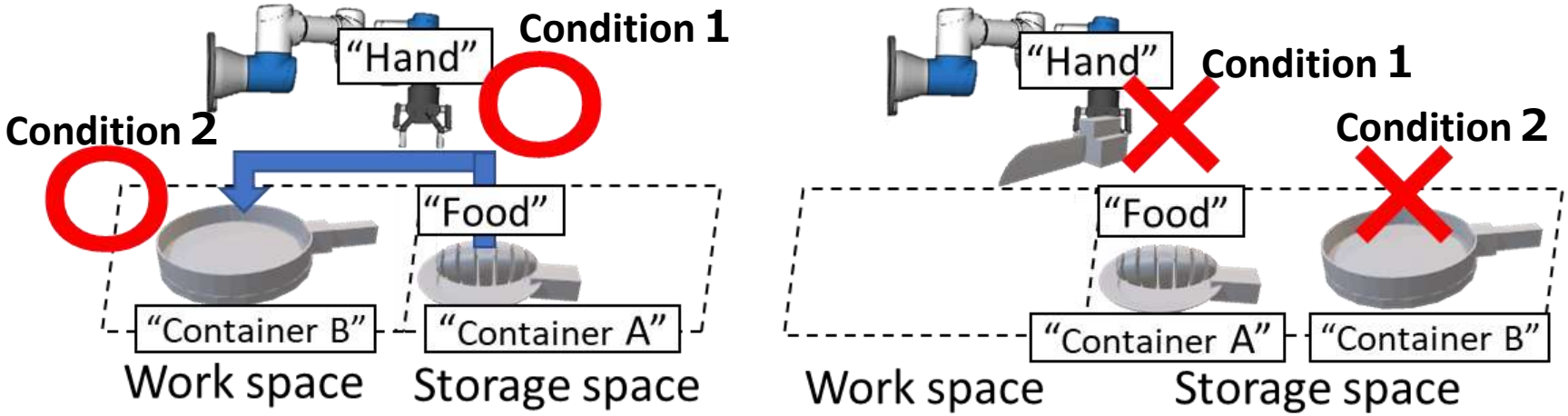
ARADA LAB
Robotic Manipulation

Executable Task Sequence

Graph Connection Based on the Task Conditions

Move “Food” from “Container A” to “Container B”

- Condition 1 “Hand” is not grasping
- Condition 2 “Container B” exists in “Work space”



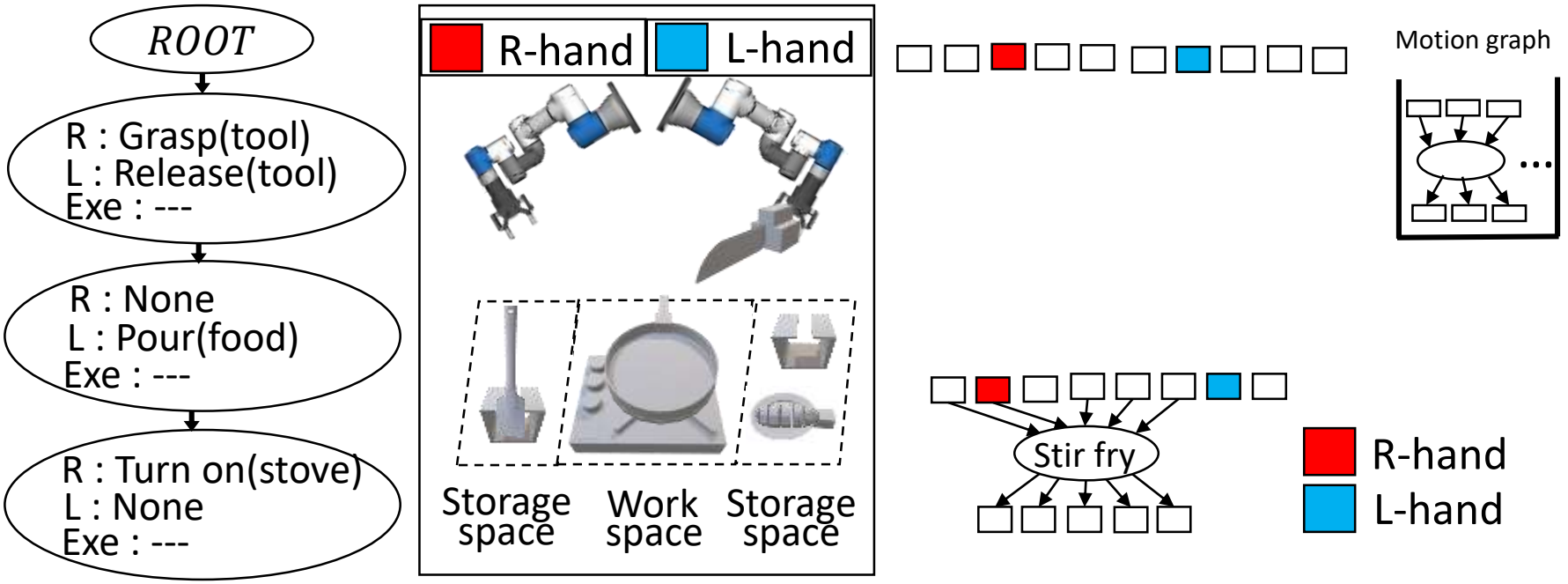
ARADA LAB
Robotic Manipulation

Executable Task Sequence

All the graph elements are connected
➡ Executable

Unexecutable Case

Task R = {Turn on(stove), Grasp(tool)}
Task L = {Release(tool), Pour(food)}



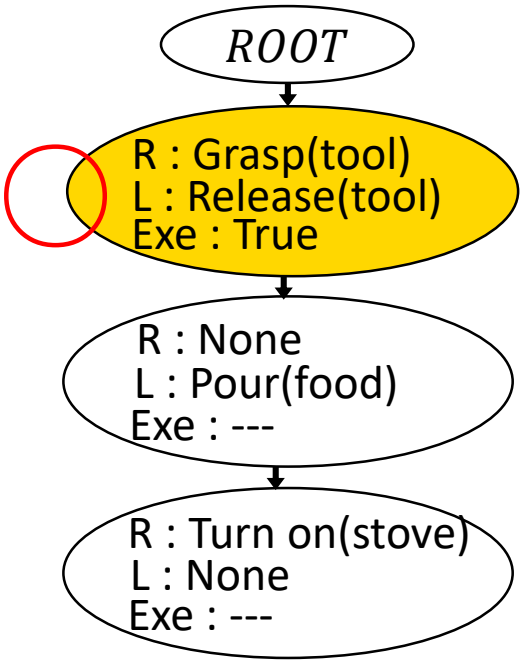
ARADA LAB
Robotic Manipulation

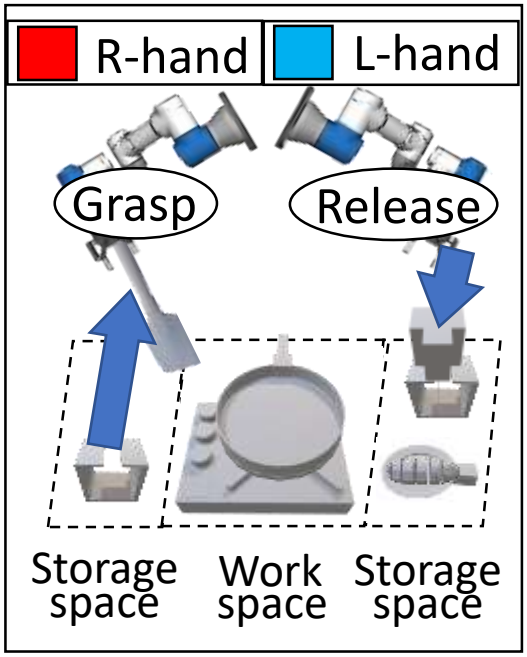
Task Executable

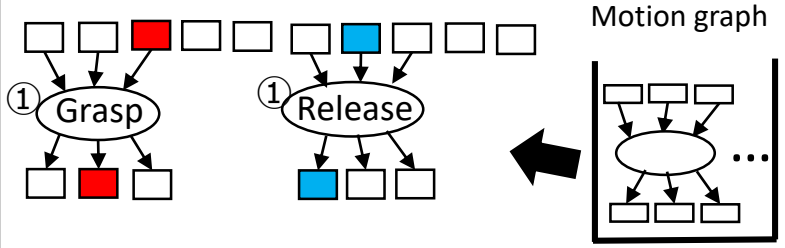
All the graph elements are connected
➡ Executable

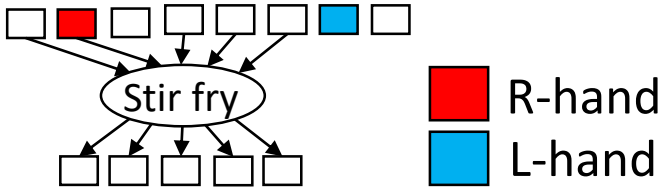
Unexecutable Case

Task R = {Turn on(stove), Grasp(tool)}
Task L = {Release(tool), Pour(food)}









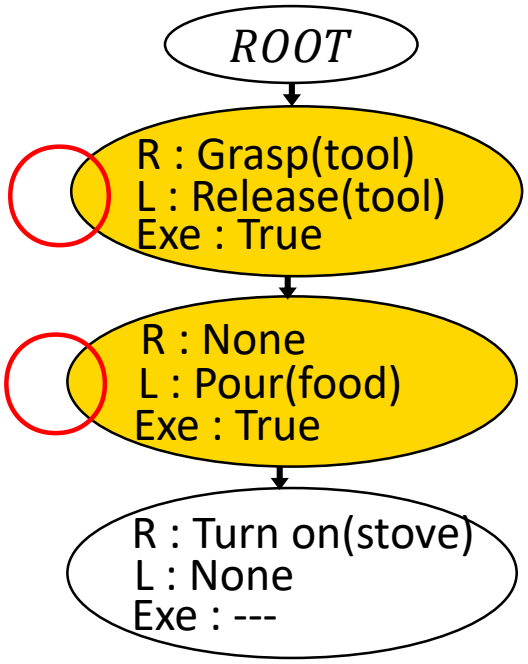
ARADA LAB
Robotic Manipulation

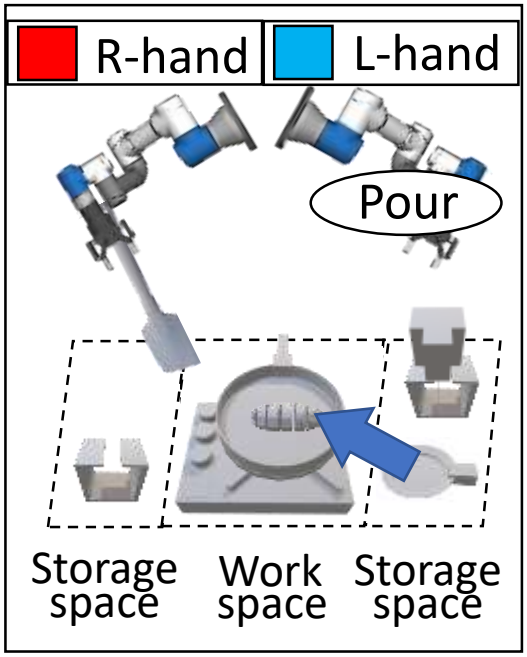
Task Executable

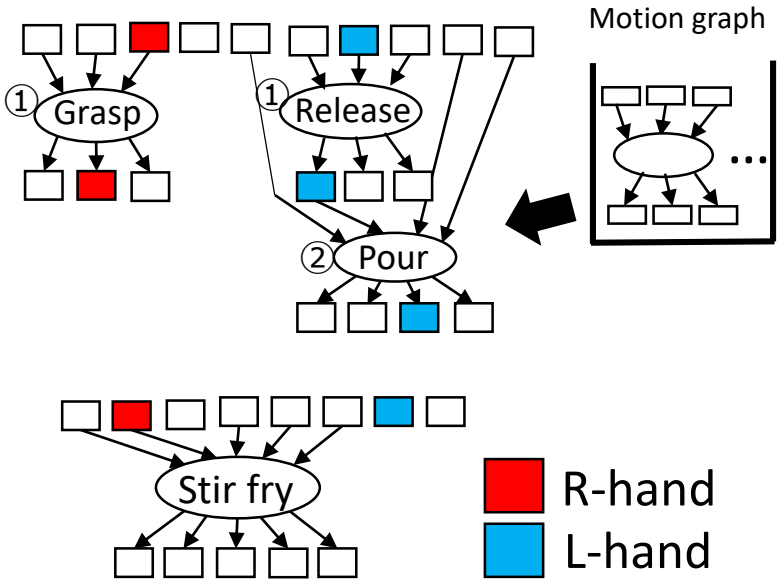
All the graph elements are connected
➡ Executable

Unexecutable Case

Task R = {Turn on(stove), Grasp(tool)}
Task L = {Release(tool), Pour(food)}







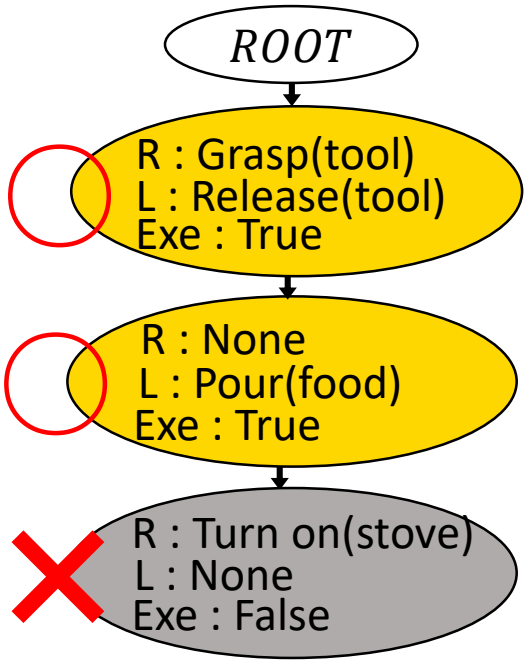
ARADA LAB
Robotic Manipulation

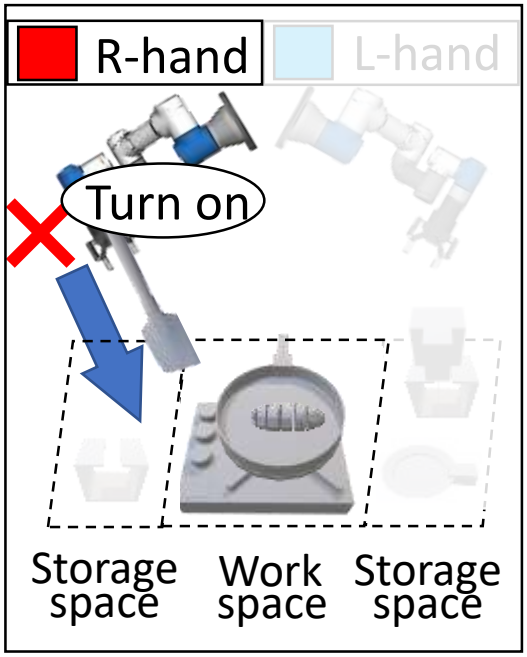
Task Executable

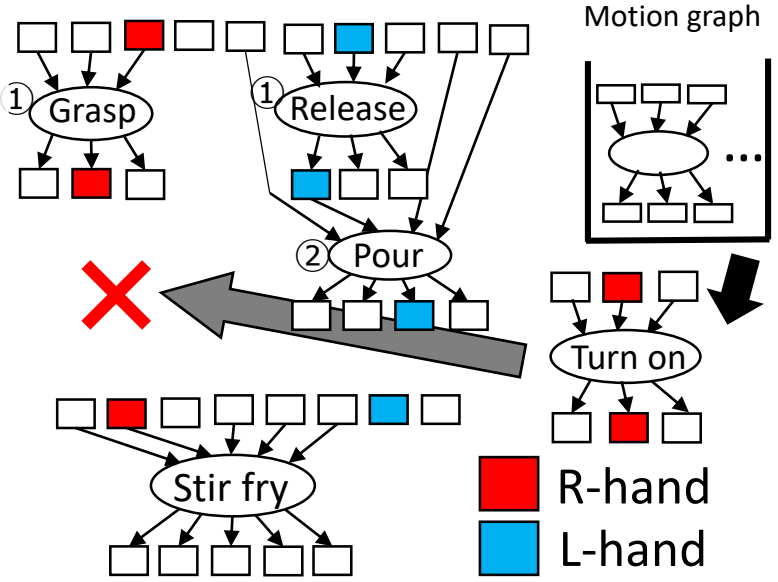
All the graph elements are connected
➡ Executable

Unexecutable Case

Task R = {Turn on(stove), Grasp(tool)}
Task L = {Release(tool), Pour(food)}







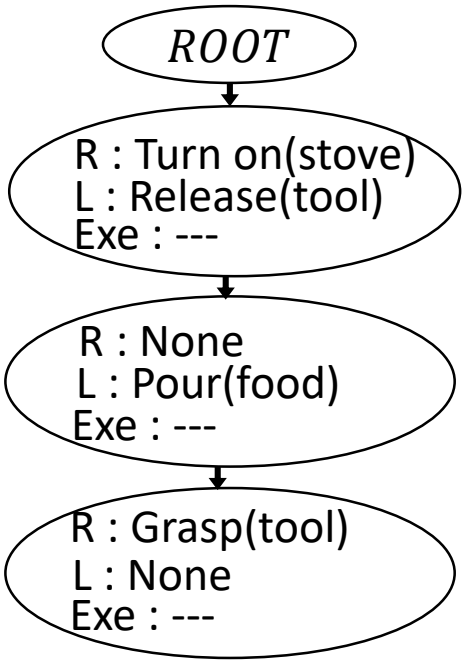
ARADA LAB
Robotic Manipulation

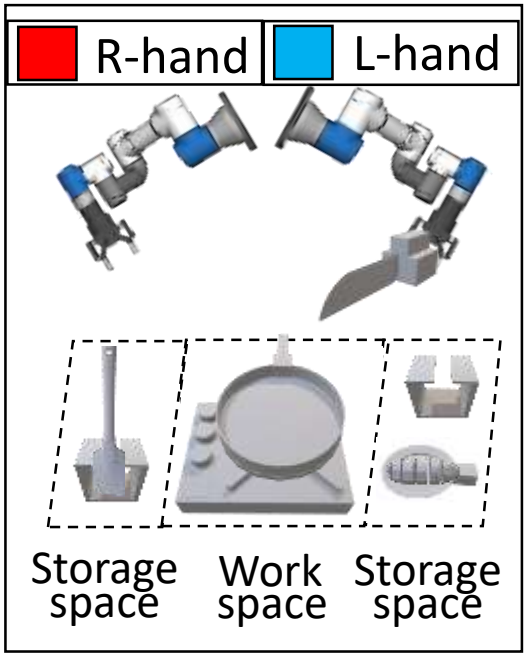
Task Executable

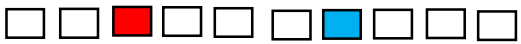
All the graph elements are connected
➡ Executable

Executable Example

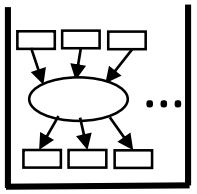
Task R = {Turn on(stove), Grasp(tool)}
Task L = {Release(tool), Pour(food)}

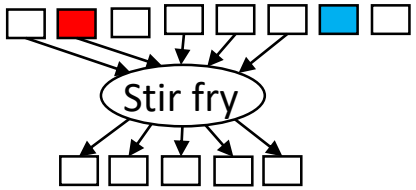






Motion graph





R-hand
L-hand

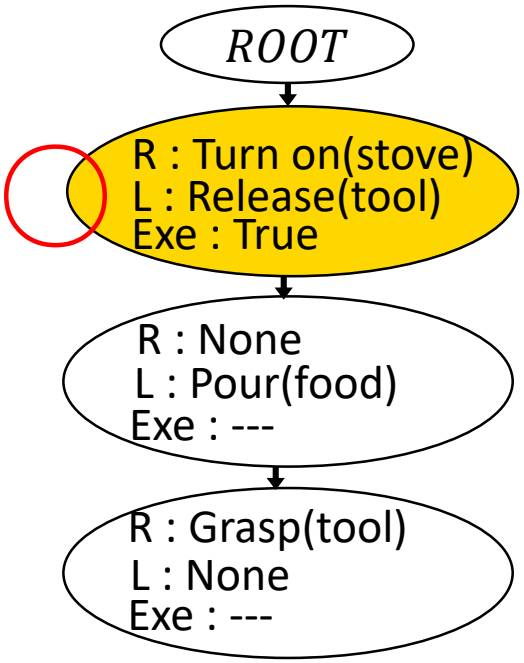
ARADA LAB
Robotic Manipulation

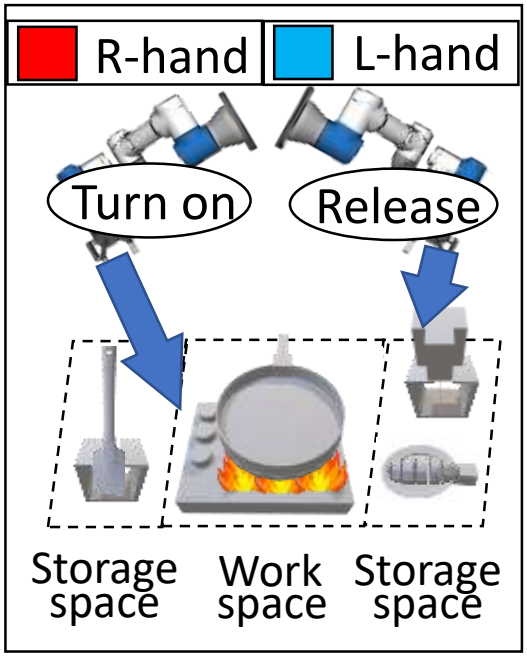
Task Executable

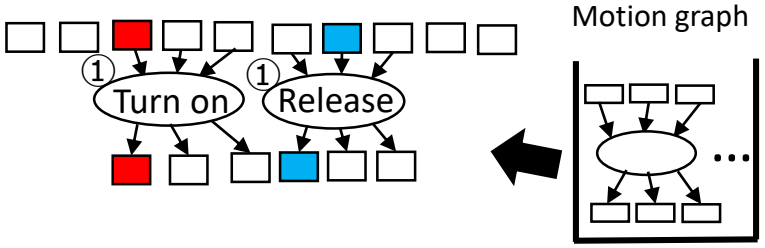
All the graph elements are connected
➡ Executable

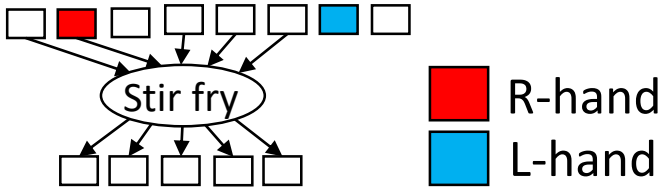
Executable Example

Task R = {Turn on(stove), Grasp(tool)}
Task L = {Release(tool), Pour(food)}









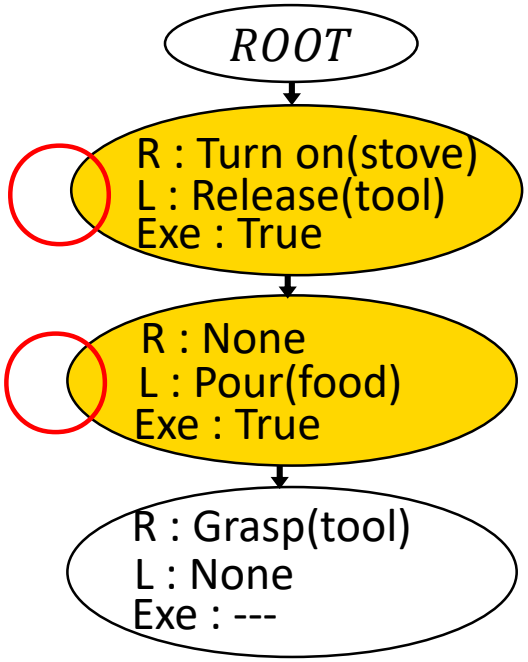
ARADA LAB
Robotic Manipulation

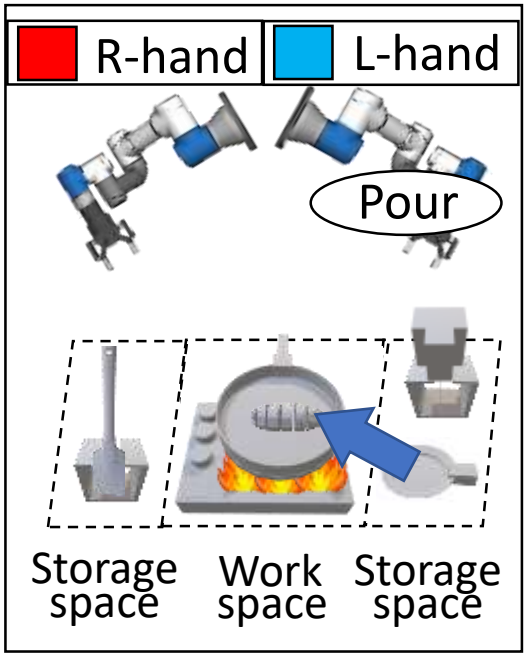
Task Executable

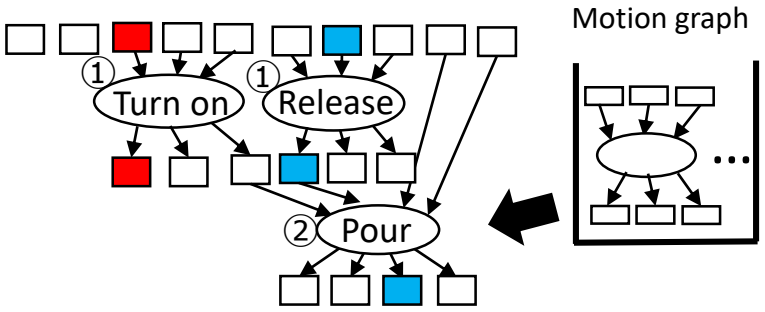
All the graph elements are connected
➡ Executable

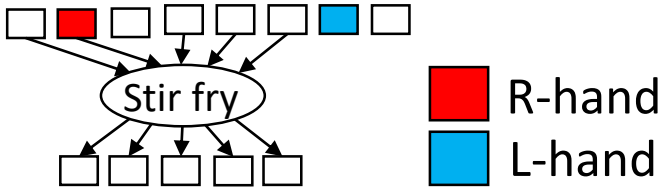
Executable Example

Task R = {Turn on(stove), Grasp(tool)}
Task L = {Release(tool), Pour(food)}









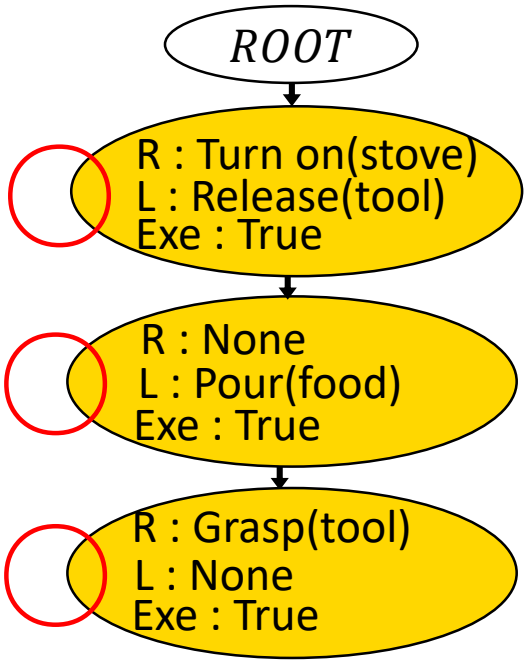
ARADA LAB
Robotic Manipulation

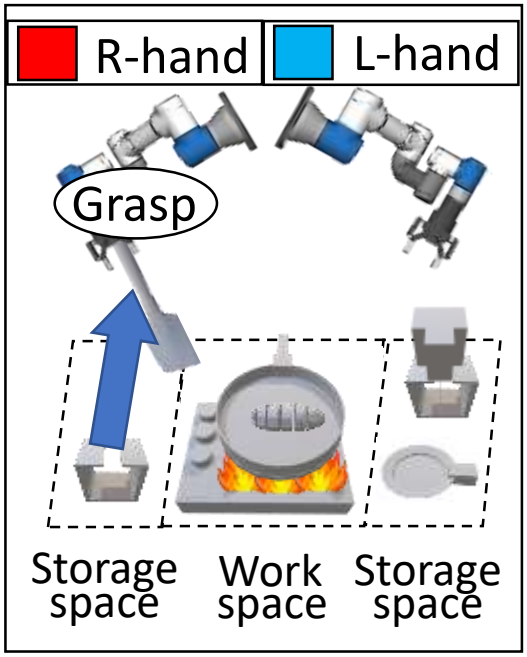
Task Executable

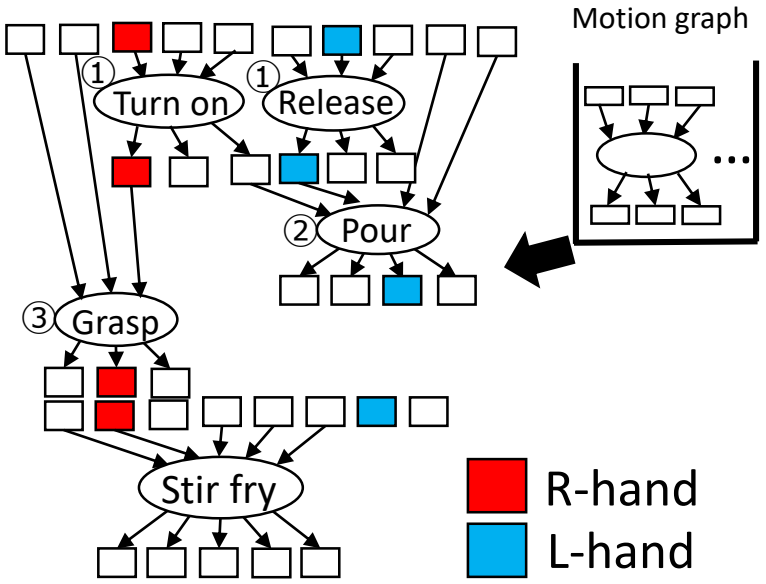
All the graph elements are connected
➡ Executable

Executable Example

Task R = {Turn on(stove), Grasp(tool)}
Task L = {Release(tool), Pour(food)}







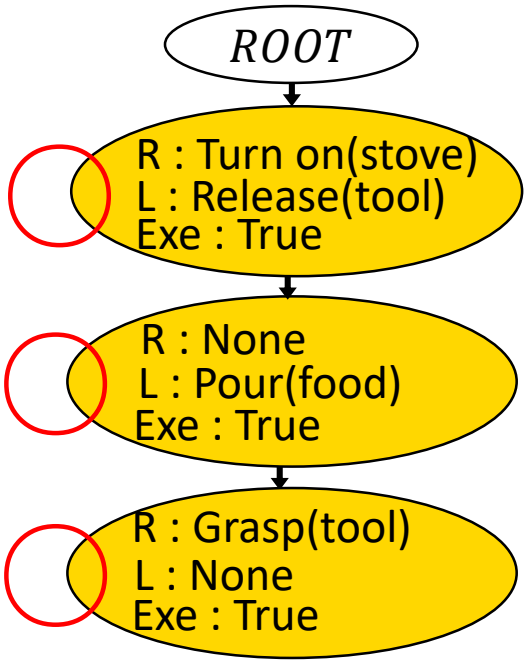
ARADA LAB
Robotic Manipulation

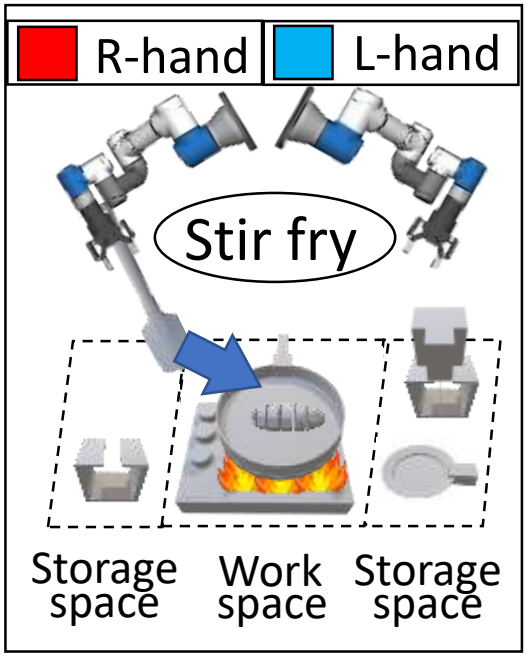
Task Executable

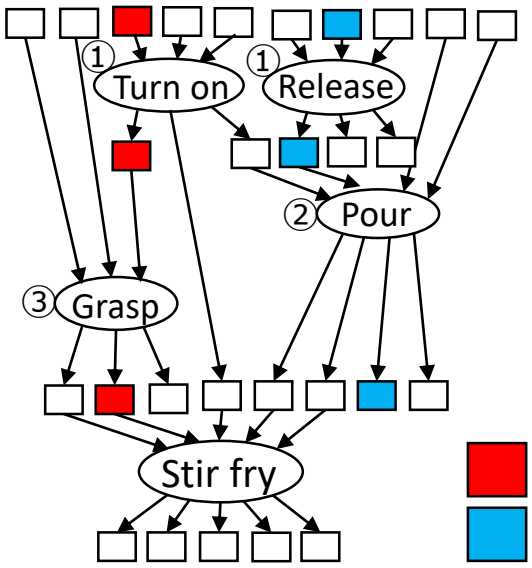
All the graph elements are connected
➡ Executable

Executable Example

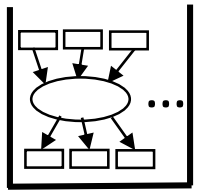
Task R = {Turn on(stove), Grasp(tool)}
Task L = {Release(tool), Pour(food)}







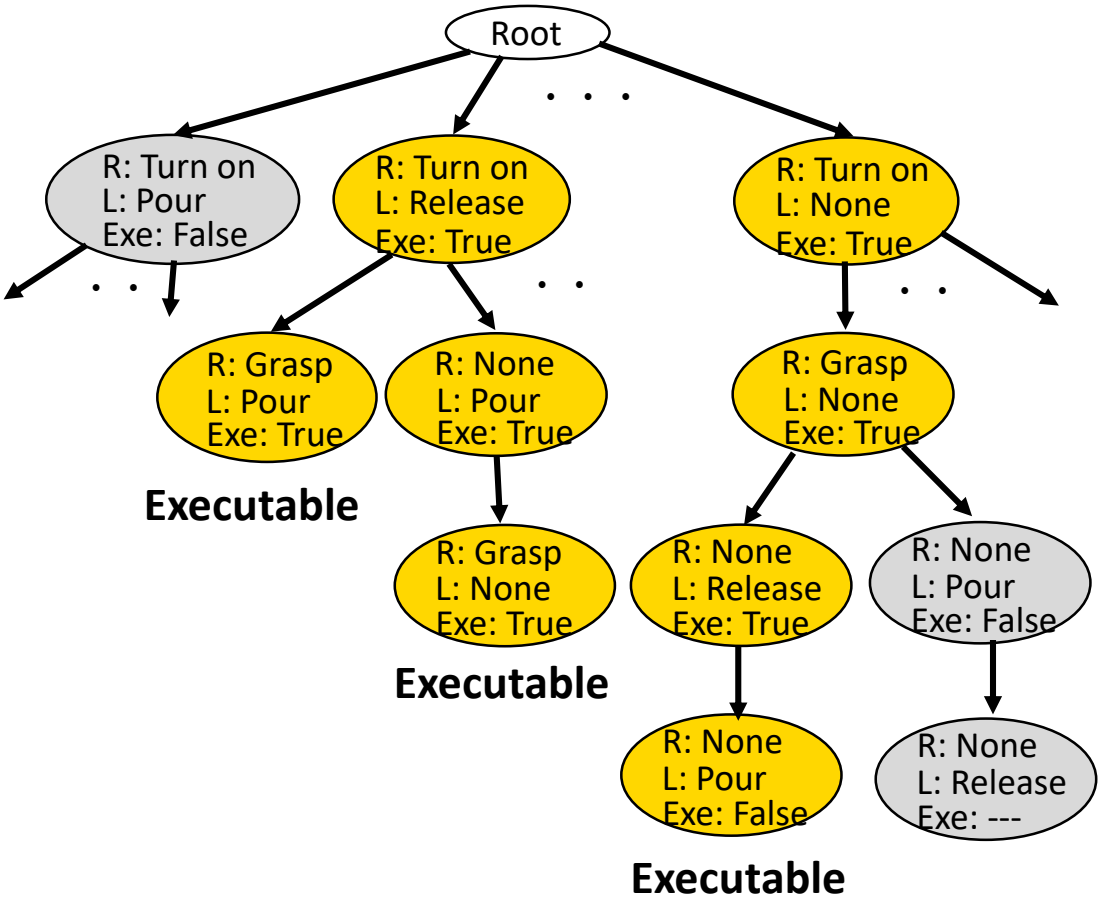
Motion graph



ARADA LAB
Robotic Manipulation

Determination of Motion Sequence

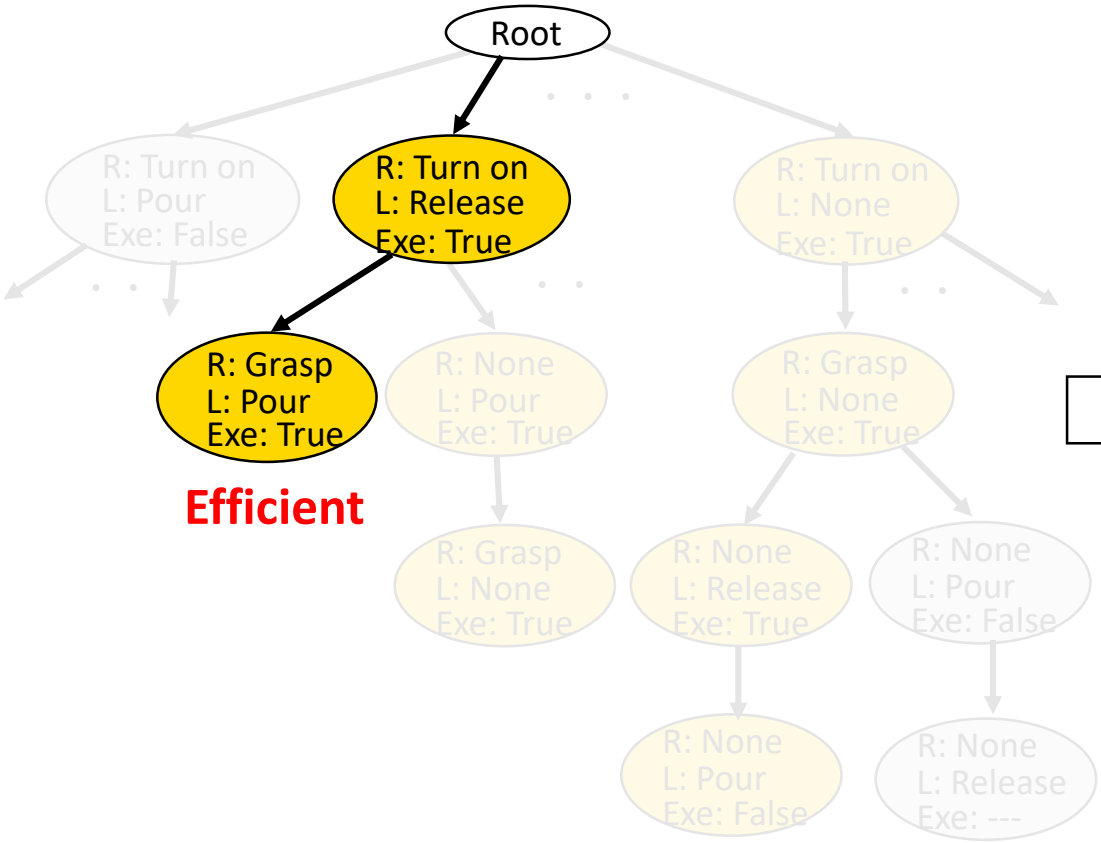
Check executable for all possible cases

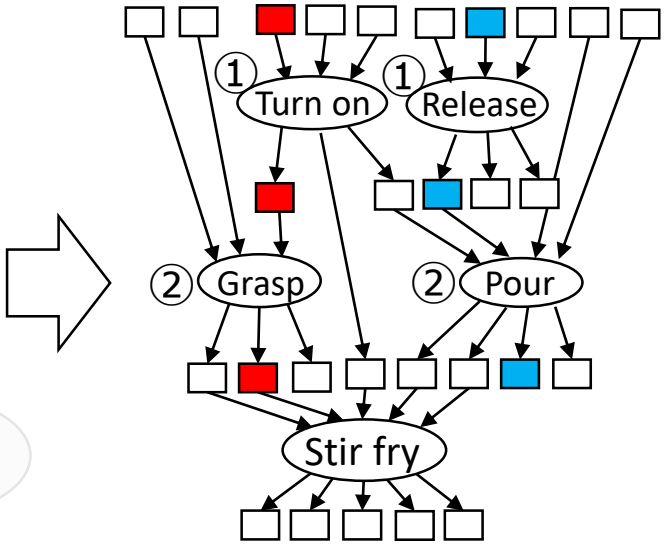


ARADA LAB
Robotic Manipulation

Determination of Efficient Motion Sequence

Least Number of Sequence





R-hand
L-hand

Method Overview

ARADA LAB
Robotic Manipulation

1st Step

Information Completion from Food Image

Estimation of Food State

2nd Step

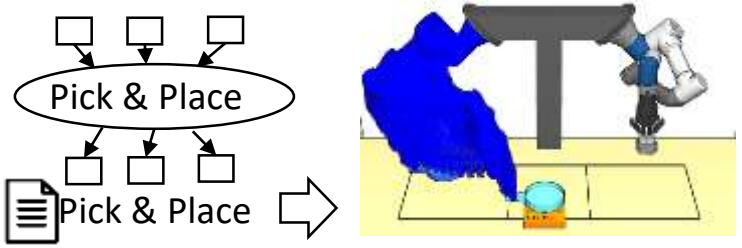
Information Completion of What is not Explicitly Written in Recipe

Comparison of Input/Output of Graph Elements

3rd Step

Motion Planning

by using Information obtain from the Graph (e.g., pick and place)



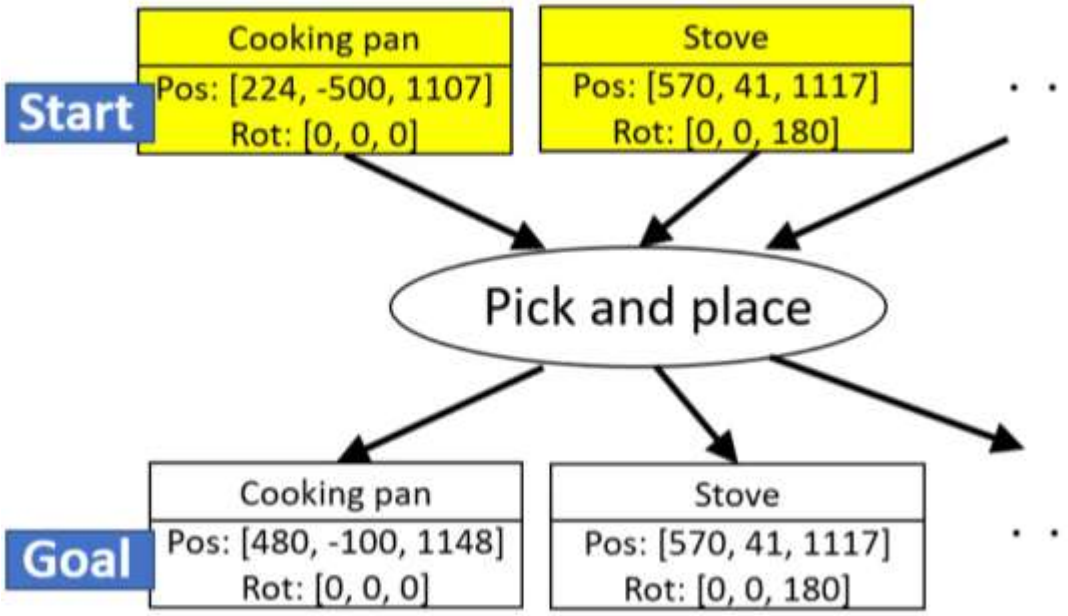
Connection of Graph Elements for Efficient Bimanual Task Planning

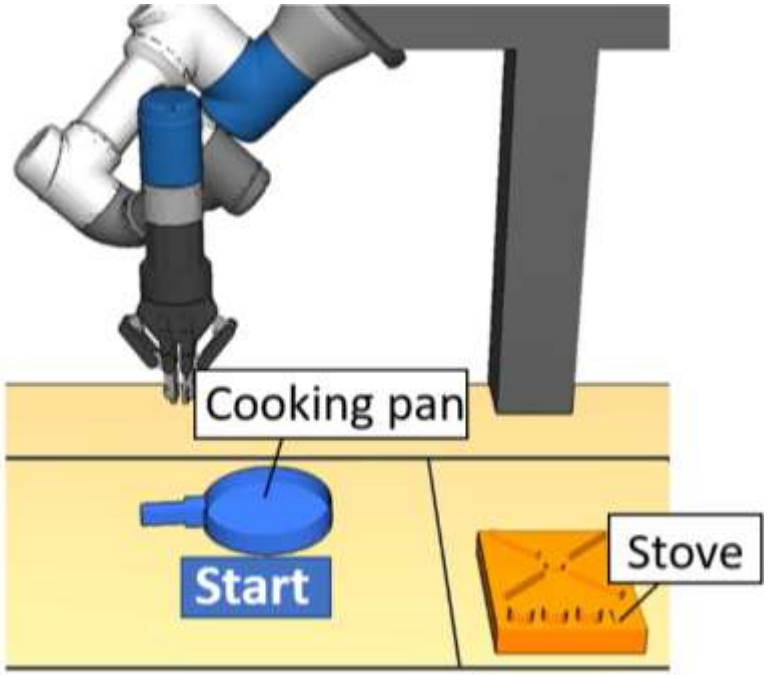
Motion Planning from Graph Information

Use Position/orientation Information in Graph Nodes

E.g., Moving “Cooking pan” to the top of “Stove”.

Determine the position of “Cooking pan” from the center of “Stove”



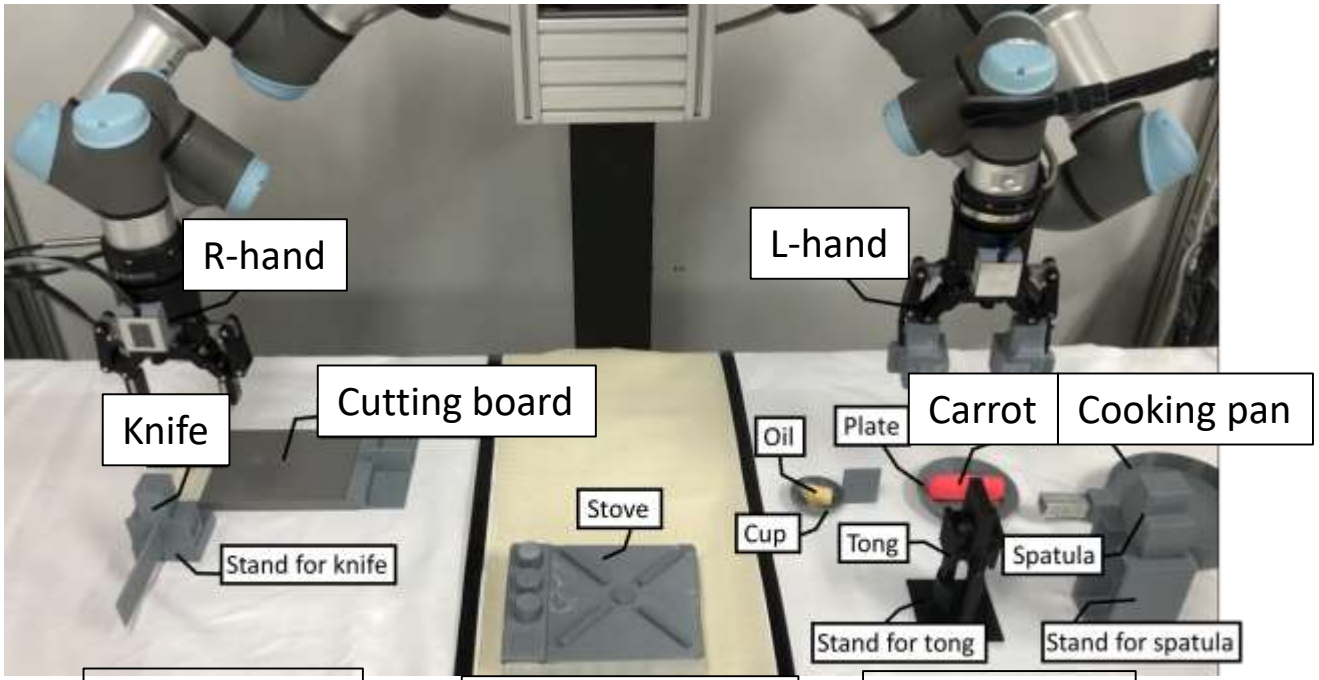


Experimental Setup

■ Recipe

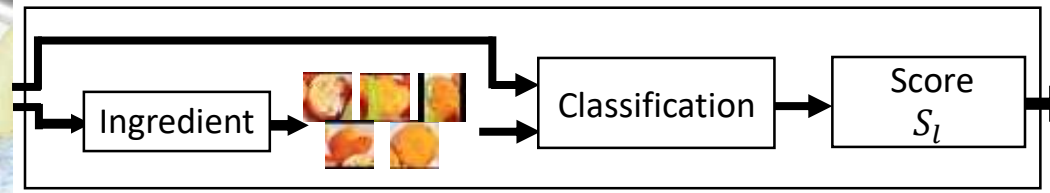
1. Heat a frypan over high heat
2. Cut a carrot in an adequate size. (Vague instruction)
3. Saute the carrot over high heat with the frypan.

■ Experimental Environment and Initial Setting



ARADA LAB
Robotic Manipulation

■ Information Completion from Food Image



Julienne	0.0
Random	0.25
Slice	**2.90**

状態スコア S_l

2. Cut a carrot into adequate size. → 2. Cut a carrot into slice.

■ Information Completion of What is not written in Recipe

For 3 instructions, 11 Implicit motion is necessary

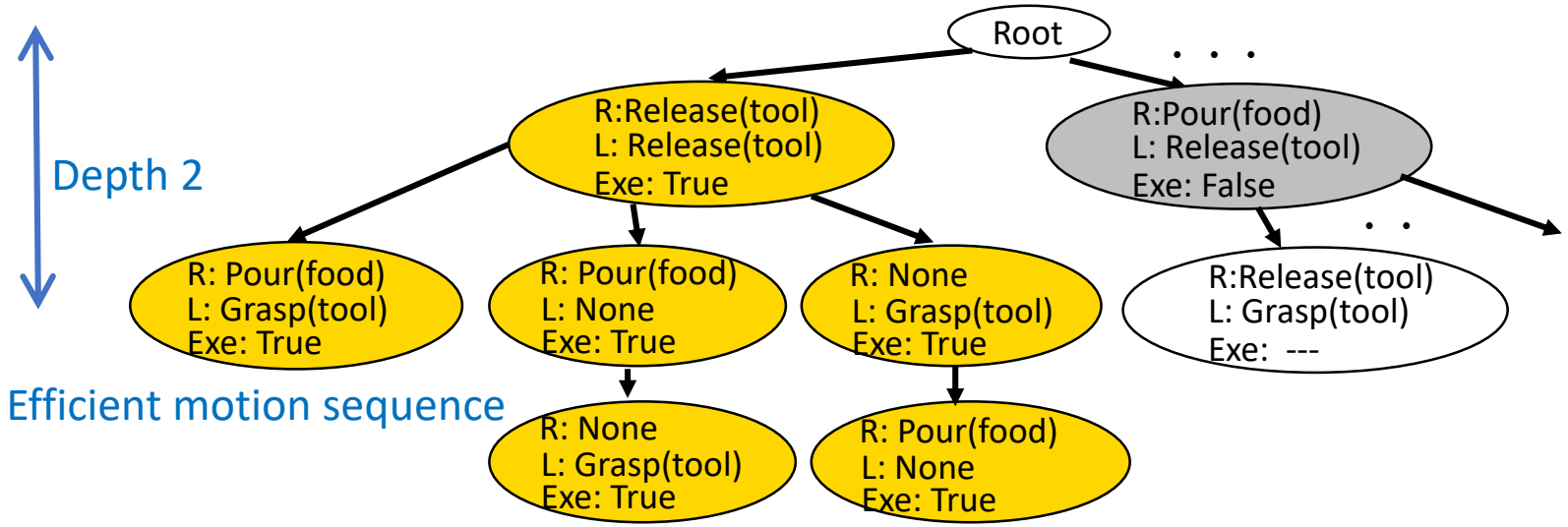
	Task R	Task L	Task RL
Instruct 1. Heat		Pick and place (container) Pour (food)	Turn on (equipment)
Instruct 2. Cut	Grasp (tool) Pick and place (container)	Grasp (tool) Pick and place (food)	
Instruct3. Stir fry	Release (tool)	Release (tool) Grasp (tool)	Pour (food)

ARADA LAB
Robotic Manipulation

■ Information Completion of What is not written in recipe

	Implicit motion	Candidate motion sequence	Feasible motion sequence	Feasible tree depth
Instruct 1. Heat	3	16	2	Min:3 Max:3 Ave:3.00
Instruct 2. Cut	4	52	5	Min:3 Max:4 Ave:3.60
Instruct3. Stir fry	4	94	20	Min:2 Max:4 Ave:3.45

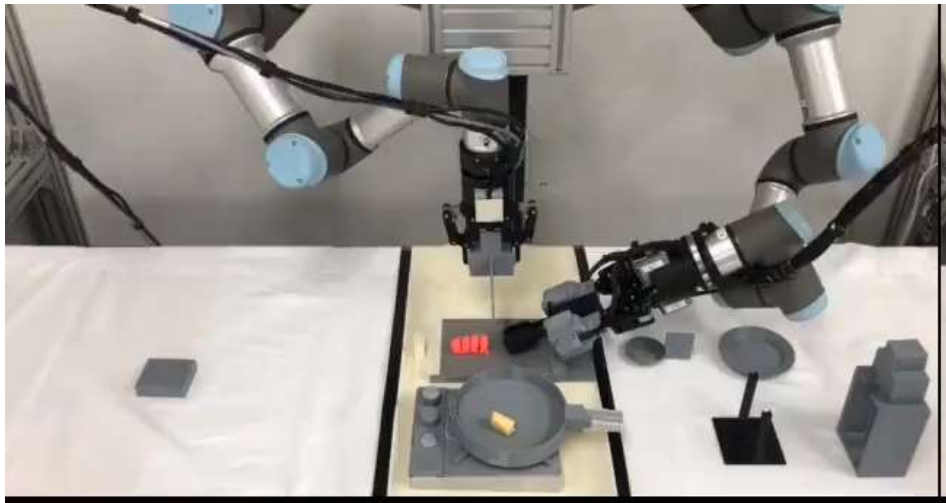
Tree Structure of Instruct 3



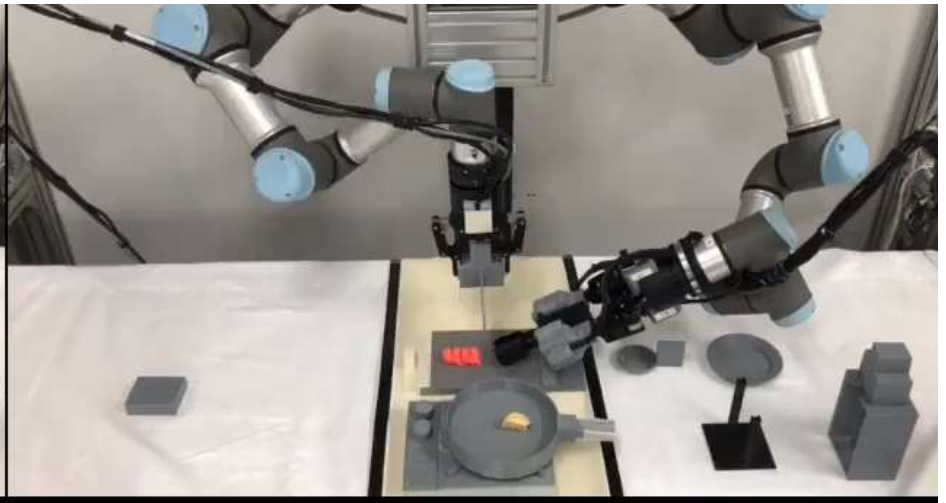
Task Efficiency by using Two-arms

ARADA LAB
Robotic Manipulation

×16



Dual-arm



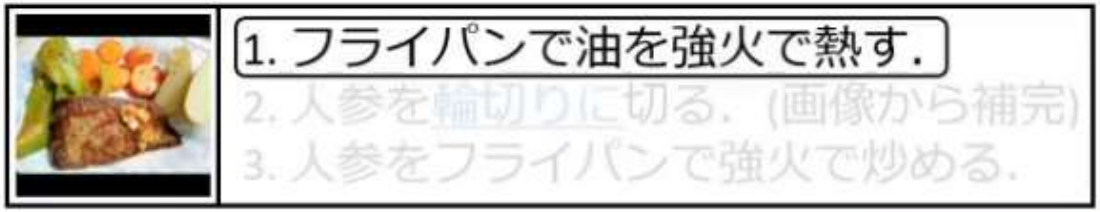
Single-arm

Right arm	Left arm
Release (tool)	Release (tool)
Pour (food)	Grasp (tool)

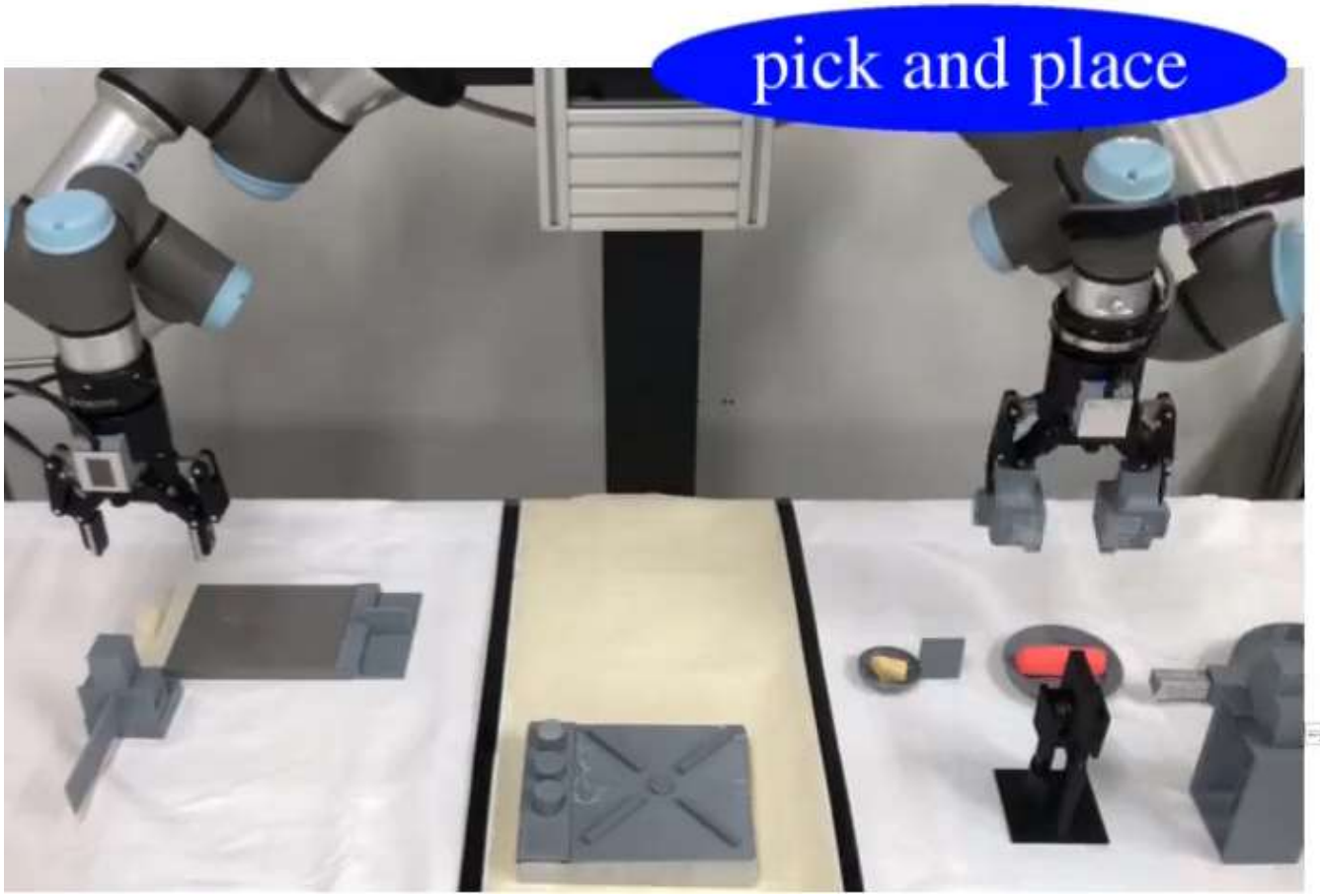
Right arm	Left arm
Release (tool)	--
--	Release (tool)
Pour (food)	--
--	Grasp (tool)

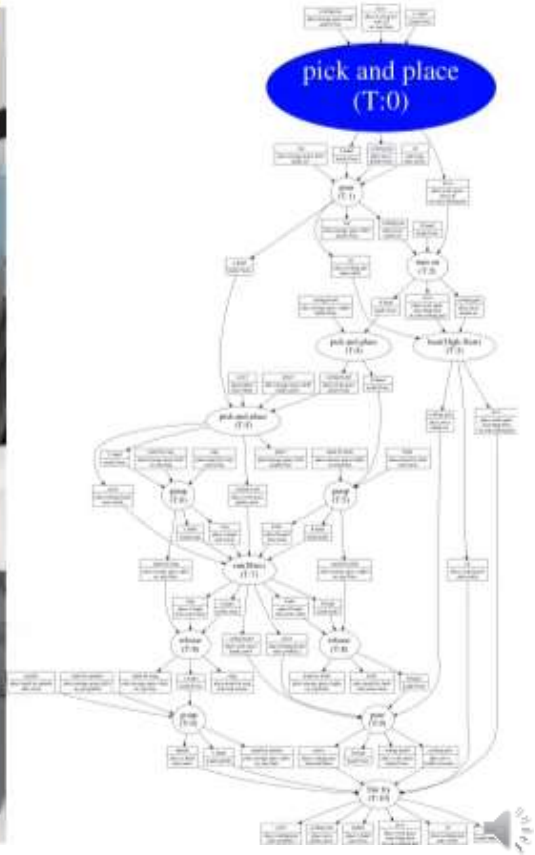
Dual-arm Manipulation Planning from Cooking Recipe

ARADA LAB
Robotic Manipulation



1. Heat a frypan
2. Cut a carrot
3. Saute the carrot over high heat





Conclusions

ARADA LAB
Robotic Manipulation

Dual-arm manipulation planning from cooking recipe based on verbal instruction and cooking image

- Information completion from food image
  - Estimator based on CNN
- Information completion of what is not written in recipe
  - Determine implicit motion based on the graph structure
  - Determine bimanual motion sequence
- Motion planning
  - Use information stored in the graph

Future Works

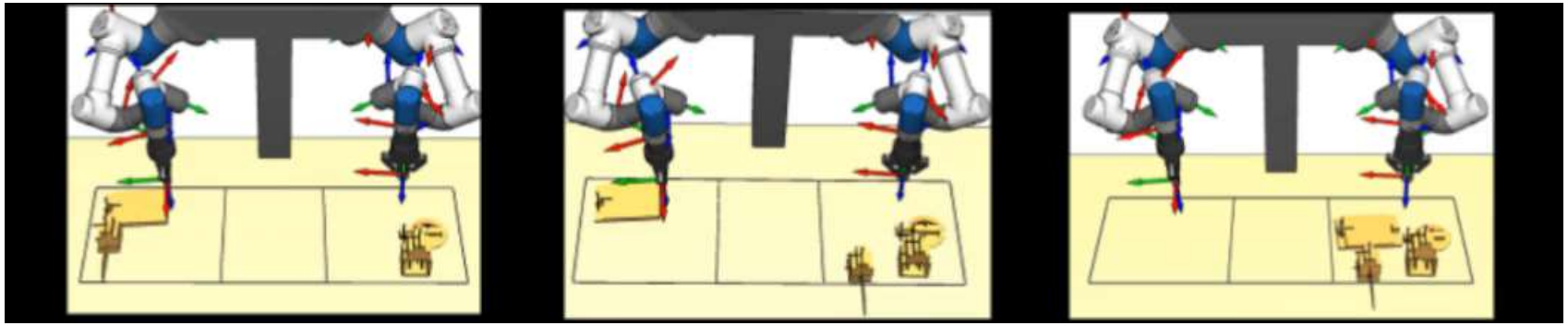
- Information completion from food image
  - Increase the variety of motion
- Information completion of what is not written in recipe
  - Validation in various cooking tasks
  - Cooperation between human and robot

様々な初期状態における作業計画

ARADA LAB
Robotic Manipulation

下記のシチュエーションでも作業動作計画の生成が可能.
しかし動作生成の成否は，対象物の配置位置次第.

指示：人参を輪切りに切る.



T	Right arm	Left arm
0	Pick&Place (cutboard)	---
1	Grasp (Knife)	Pick&Place (Carrot)
2	---	Grasp (Tong)
3	Cut(slice)	Cut(slice)

T	Right arm	Left arm
0	Pick&Place (Cutboard)	Grasp (Knife)
1	Handover (Knife)	Handover (Knife)
2	---	Pick&Place (Carrot)
3	---	Grasp (Tong)
4	Cut(slice)	Cut(slice)

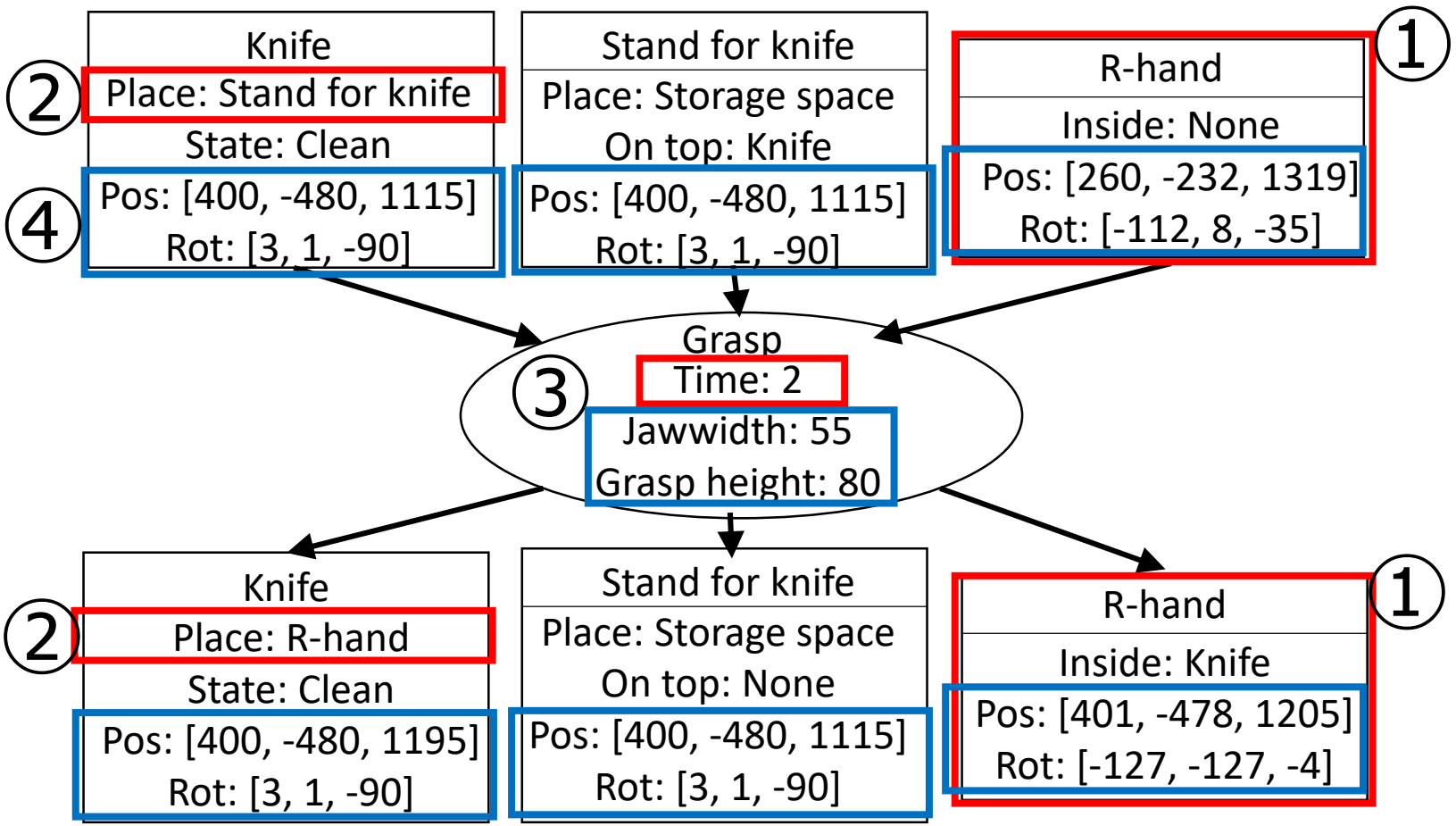
T	Right arm	Left arm
0	---	Pick&Place (Cutboard)
1	---	Pick&Place (Carrot)
2	---	Grasp (Knife)
3	Handover (Knife)	Handover (Knife)
4	---	Grasp (Tong)
5	Cut(slice)	Cut(slice)

新規性

- 言語指示で欠如する情報を画像から補完
- レシピに明示されない動作を含めた作業シーケンスグラフ生成
  - 従来研究を拡張したグラフでタスクを表現
  - 双腕の並列処理を考慮した作業手順の決定

先行研究とのグラフ構造の相違点

① 動作で使用するロボットを考慮するためにハンドノードを新たに導入
② 道具や容器の存在位置をオブジェクトノードの属性に持つ
③ 動作の実行順序をモーションノードの属性に持つ
④ 各ノードで、動作計画に必要な情報を持つ(対象物の位置姿勢など)

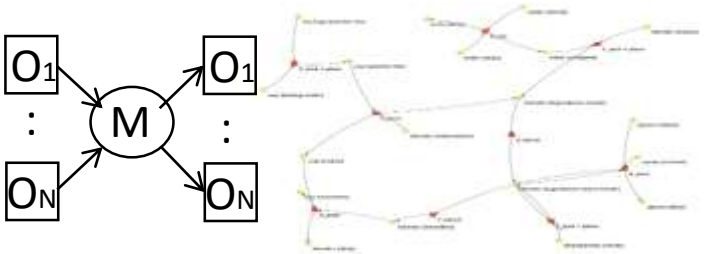


先行研究

動画を基にした料理作業計画[1][2]

料理動画

グラフネットワーク(FOON)

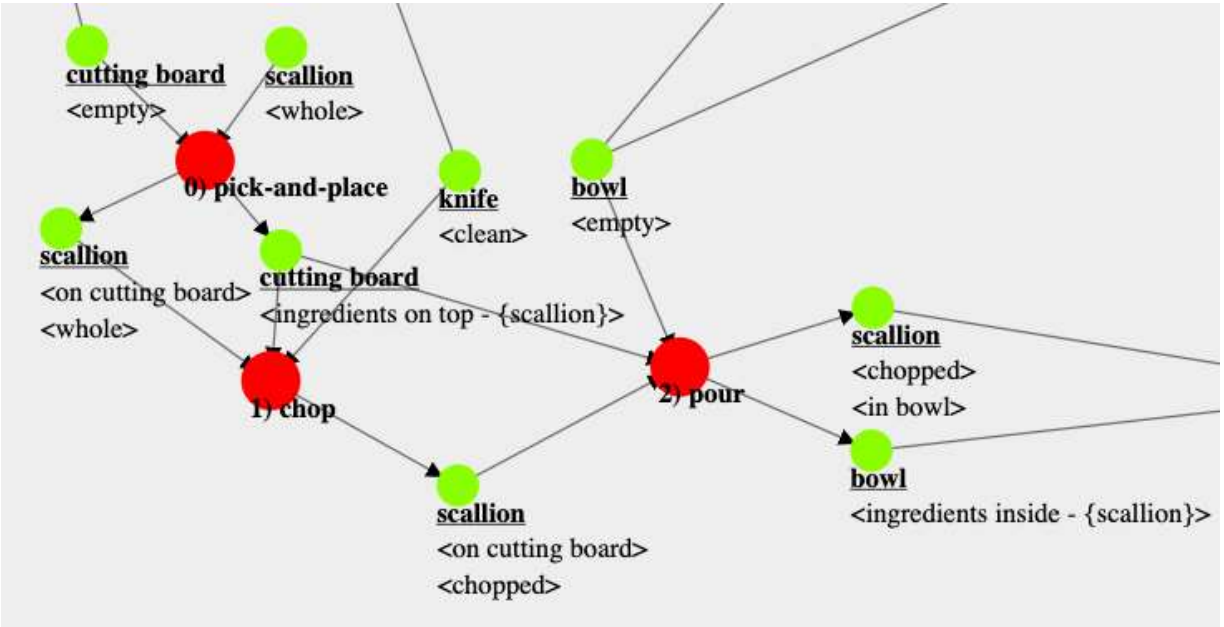


動作実行

例. 春玉ねぎを切り、ボウルに注ぐ

手順

1. Pick and place (食材→まな板)
2. Chop
3. Pour



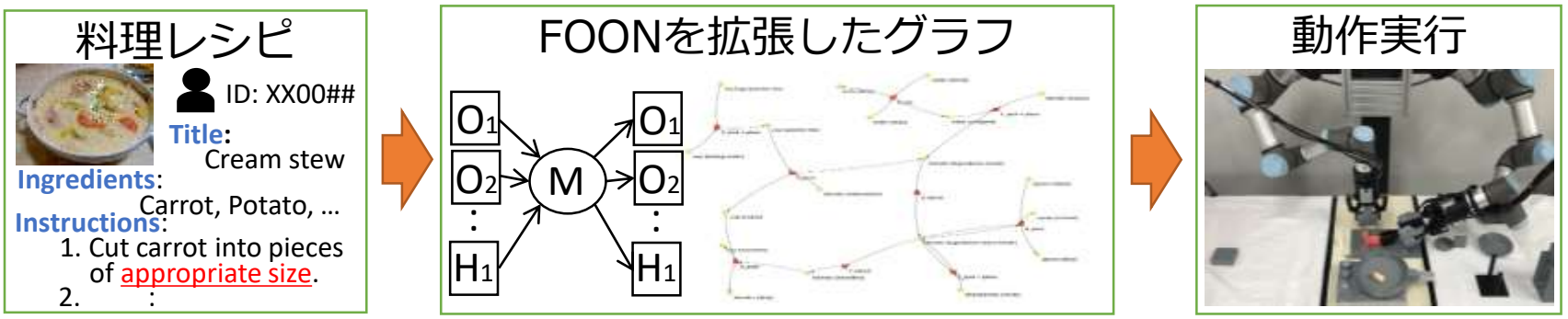
[1] PAULIUS et al. “Functional object-oriented network for manipulation learning,” IROS 2016

[2] PAULIUS et al. “A Weighted Functional Object-Oriented Network for Task Planning,” ICRA 2019

本研究

ARADA LAB
Robotic Manipulation

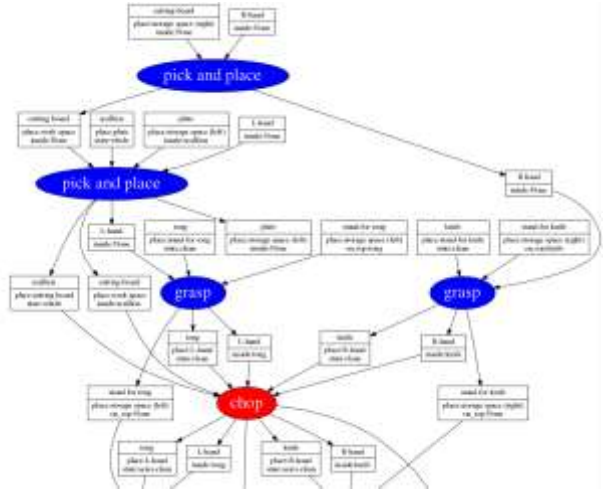
レシピを基にした料理作業**動作**計画



例. 春玉ねぎを切り、ボウルに注ぐ

手順

1. Pick and place (まな板)
2. Pick and place (食材→まな板)
3. Grasp (包丁,トング)
4. Chop
5. Release (包丁,トング)
6. Pick and place (ボウル)
7. Pour

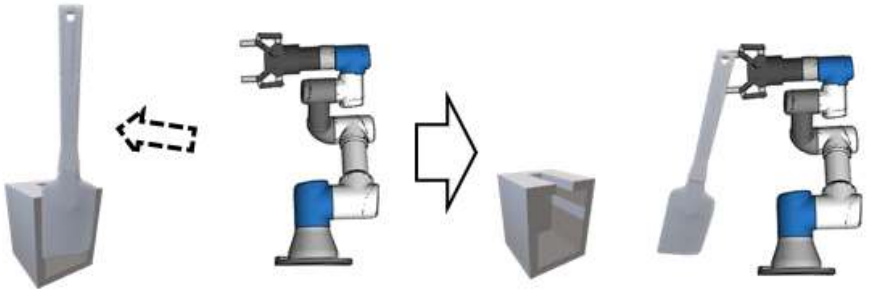


作業場の状況に応じた、料理の準備動作を考慮

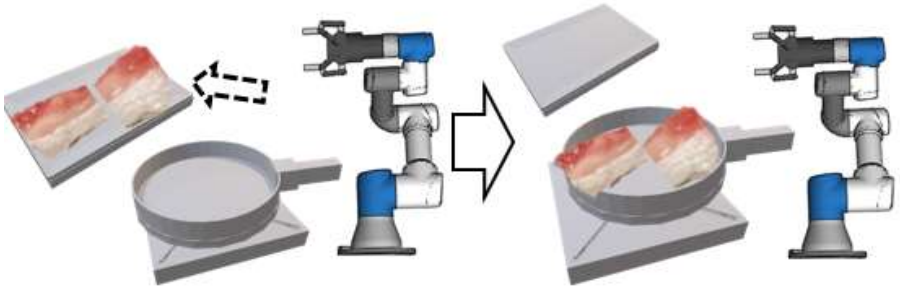
レシピに明示されない動作

ARADA LAB
Robotic Manipulation

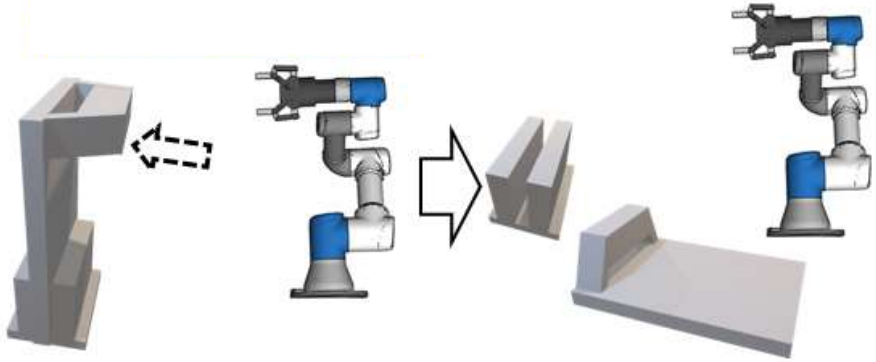
■ 道具を取ってくる
/収納場所に戻す



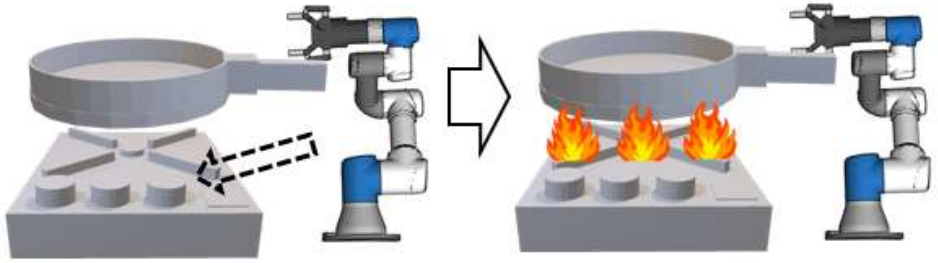
■ 食材を移動させる



■ 容器を取ってくる



■ コンロを点火する

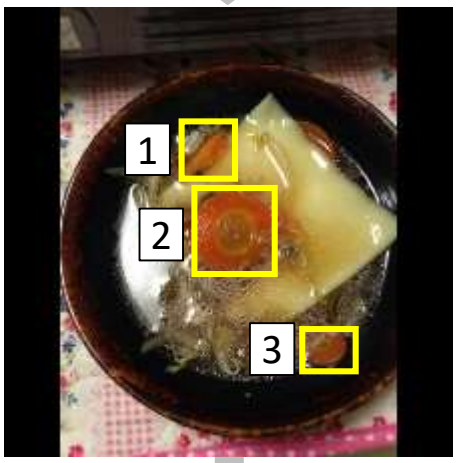


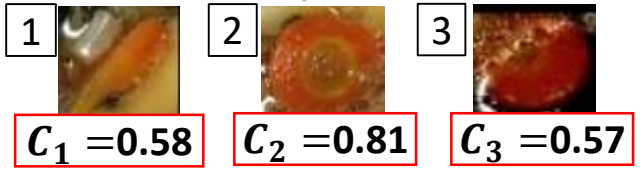
注目食材の加工状態推定システム(Model C)

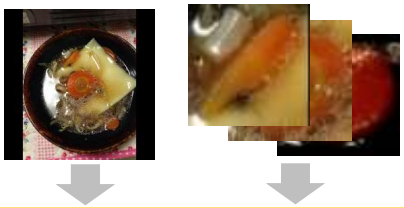
注目食材名 Carrot

料理画像

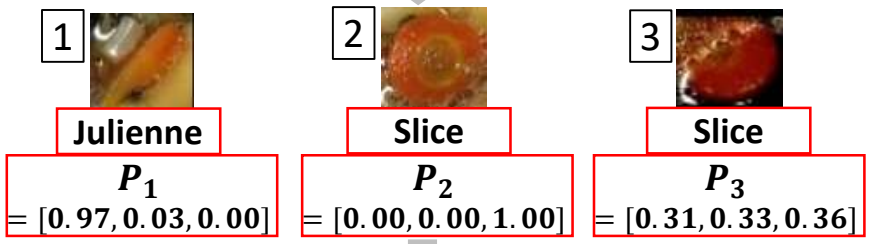
食材検出モデル
(YOLOv5[4]をFine Tuning)







状態分類モデル
(Xception[5]をFine Tuning)



$l \in \{Julienne, Random, Slice\}$

加工状態スコアS_lの算出

$$\underset{l \in L}{argmax}\, S_l \qquad S_l = \sum_{i=1}^{N} p_{i,l} C_i \qquad P_I = [\, p_{I,1}, \ldots, p_{I,M} \,] \qquad L = \{state_1, \ldots, state_M\}$$

推定結果

Slice

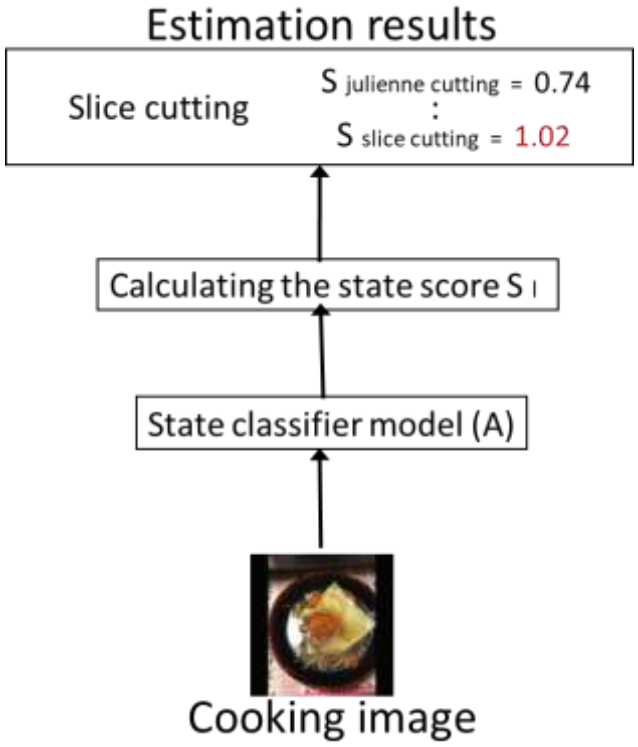
$S_{Julienne}$=0.74
S_{Random}=0.21
S_{Slice}=**1.02**

[4] Glenn Jocher; et al. Yolov5 in pytorch, 2020.https://github.com/ultralytics/yolov5
[5] Fran¸cois Chollet. Xception: Deep learning withdepthwise separable convolutions(2017)

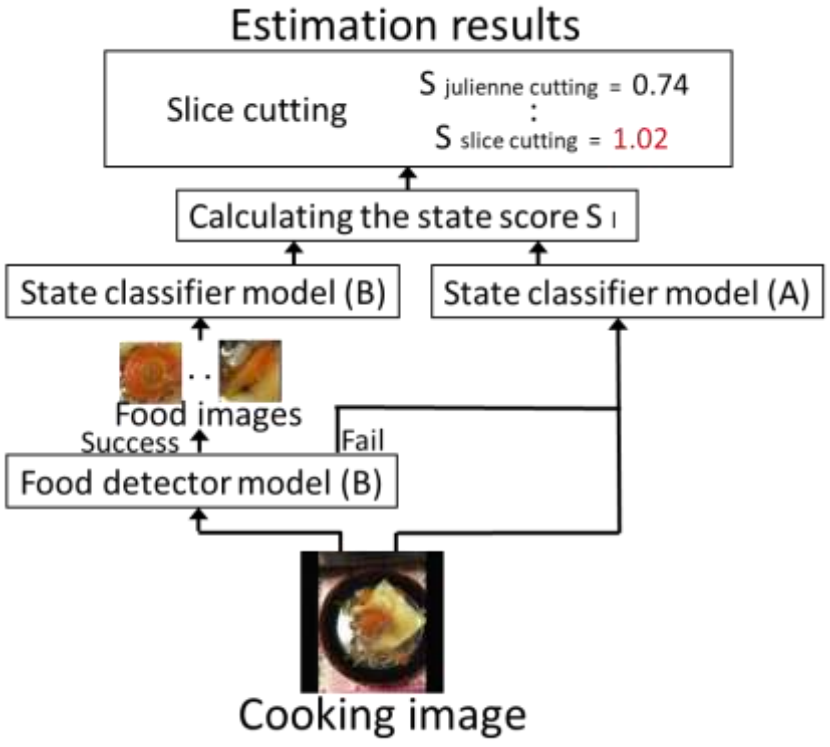
４つの状態推定モデル(1/2)

ARADA LAB
Robotic Manipulation

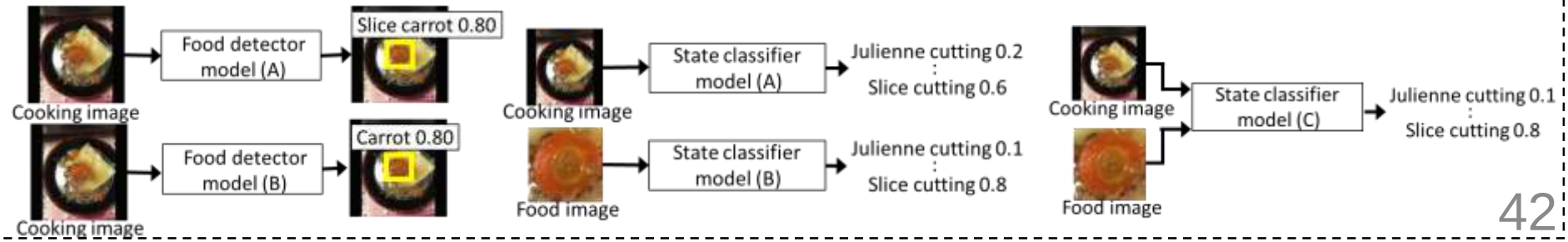
■ モデルA (料理画像を基に推定)



■ モデルB (検出された食材画像を基に推定)



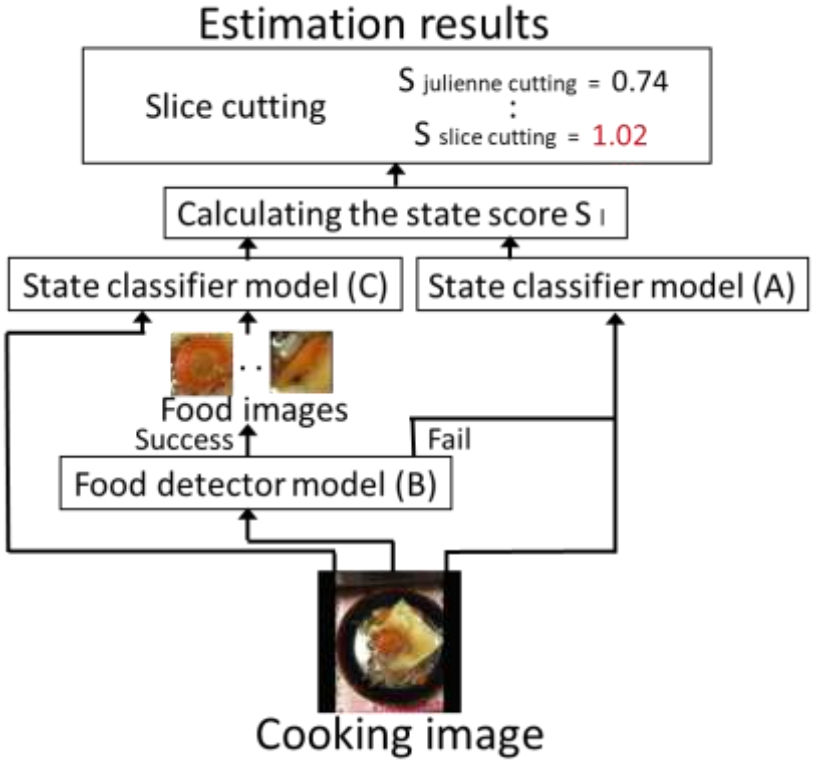
※推定モデルを構成する2種の食材検出と3種の状態分類モデル



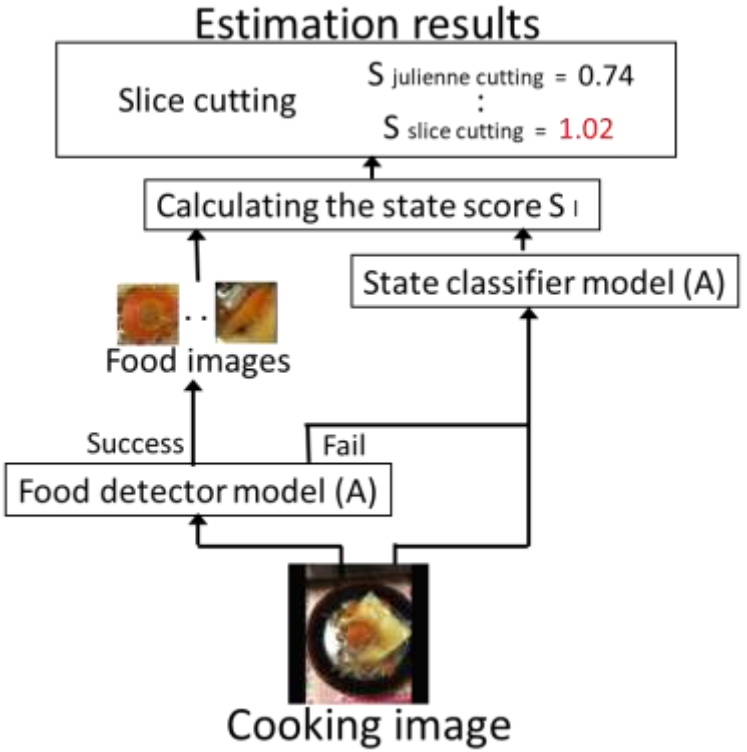
４つの状態推定モデル(2/2)

ARADA LAB
Robotic Manipulation

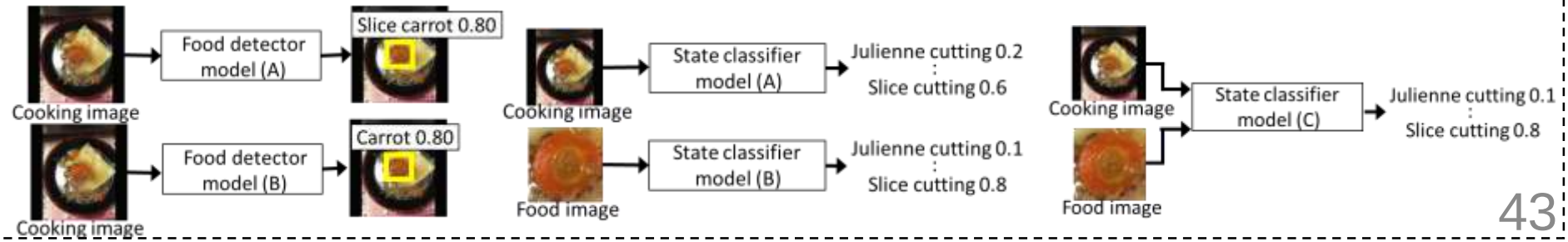
■ モデルC (検出された食材画像と料理画像を基に推定)



■ モデルD (食材検出と状態分類を同時に行い得られた結果を基に推定)



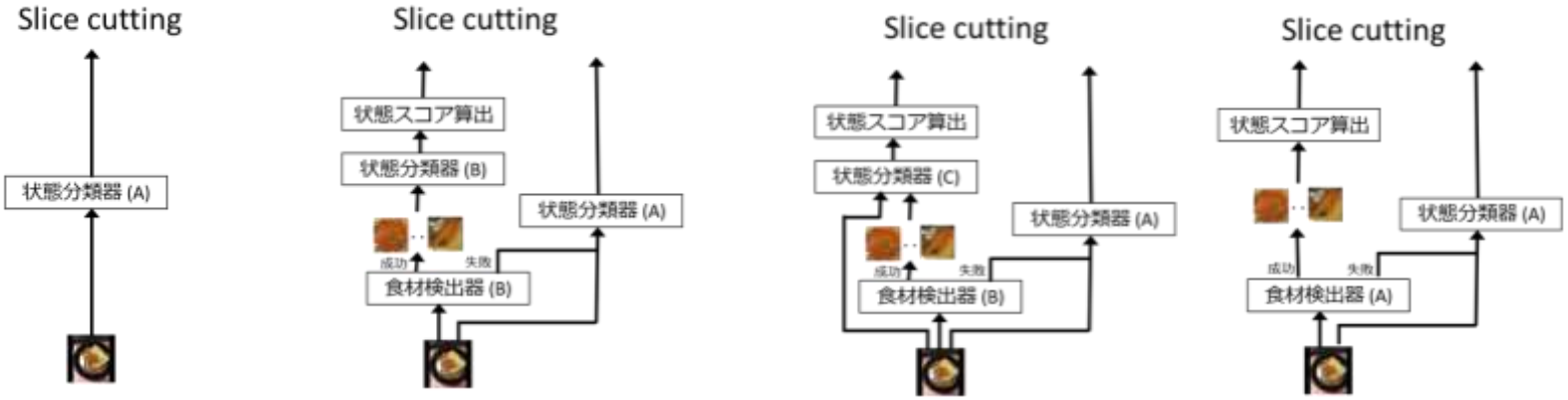
※推定モデルを構成する2種の食材検出と3種の状態分類モデル



評価方法

■ 4つの異なる構成を持つ加工状態推定システム

※カッコ内は推定の情報源



Model A (料理画像)　Model B (検出された食材画像)　Model C (検出された食材画像＋料理画像)　Model D (料理画像)

■ 1食材あたり450枚の料理画像(楽天レシピデータ)

□ 対象食材

Carrot　Green pepper

□ 対象加工状態

Julienne　Random　Slice

評価結果

Model	Sources of estimation	System configuration	Carrot	Green pepper
A	料理画像	状態分類	71	65
B	検出された食材画像	食材検出+状態分類	75	69
C	検出された食材画像 ＋料理画像	食材検出+状態分類	**78**	**73**

※評価指標:Accuracy

一方のみを考慮(Model A,B)よりも
食材画像と料理画像の両方を考慮(Model C)することで，性能が向上

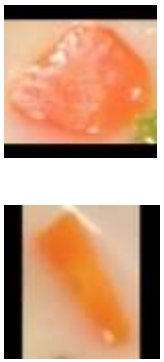
例. **Actual label : Random**

A

Predict label : Random

Julienne(細切り)	0.12
Random(乱切り)	**0.87**
Slice(輪切り)	0.01

B



Predict label : Slice

Julienne(細切り)	0.61
Random(乱切り)	0.14
Slice(輪切り)	**0.90**

C

Predict label : Random

Julienne(細切り)	0.18
Random(乱切り)	**1.25**
Slice(輪切り)	0.22

ARADA LAB
Robotic Manipulation

評価結果

Model	Sources of estimation	System configuration	Carrot	Green pepper
A	料理画像	状態分類	71	65
B	検出された食材画像	食材検出+状態分類	75	69
C	検出された食材画像 ＋料理画像	食材検出+状態分類	**78**	73
D	料理画像	食材検出	75	**74**

※評価指標:Accuracy

■ 推定の情報源

一方のみを考慮(Model A,B)よりも
食材画像と料理画像の**両方を考慮する**(Model C)ことで，性能が向上

■ 推定システムの構成

最も良い性能を持つモデル
Carrot ：食材検出と状態分類を**二段階に分ける**(Model C)
Green pepper：食材検出と状態分類を**同時に行う**(Model D)

食材の特性に応じたモデル設計が重要