Promptable Behaviors: Personalizing Multi-Objective Rewards from Human Preferences

Minyoung Hwang[1,2], Luca Weihs[1], Chanwoo Park[2], Kimin Lee[3], Ani Kembhavi[1], Kiana Ehsani[1]

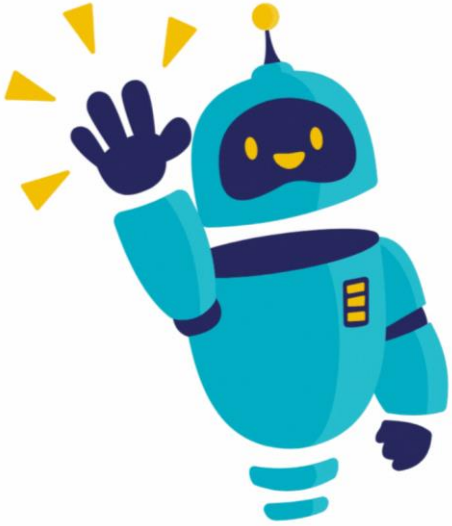
[1]Allen Institute for AI, [2]MIT, [3]KAIST

CVPR JUNE 17-21, 2024 SEATTLE, WA

Perceptual Reasoning & Interaction Research

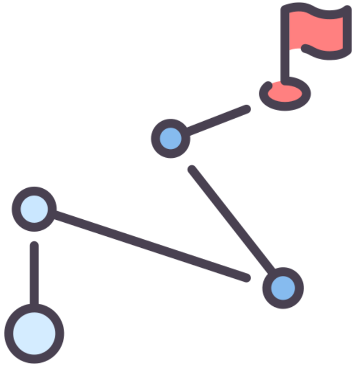
Introduction

How can we effectively customize a robot for human users?

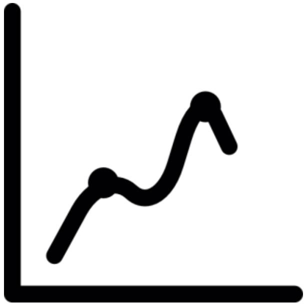


Introduction

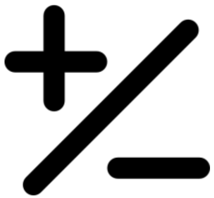
Human's objective or preference can be described in various ways.



Goal



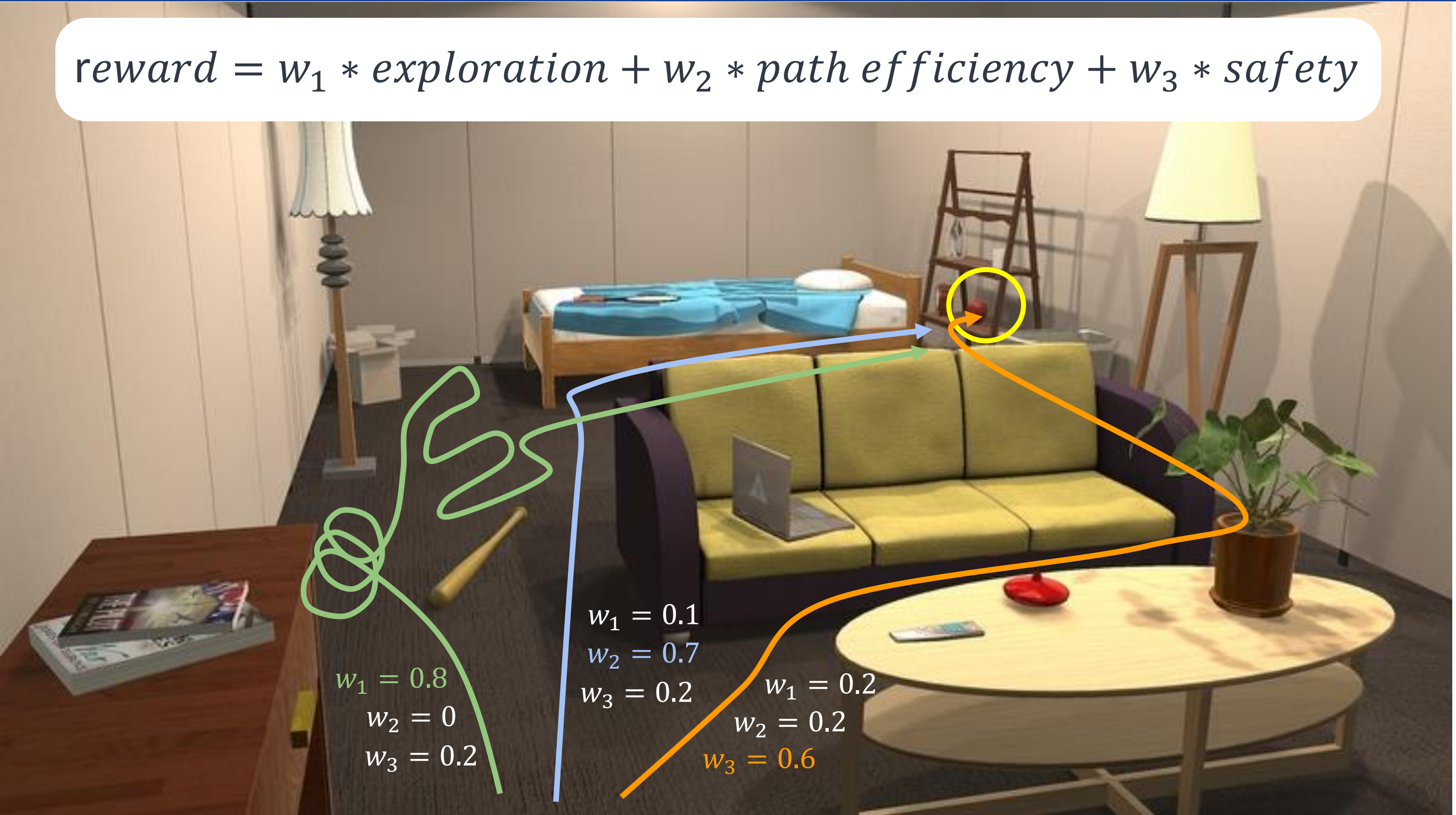
Cost Function



Reward

Introduction

$$reward = w_1 * exploration + w_2 * path\ efficiency + w_3 * safety$$



ObjectNav Task:

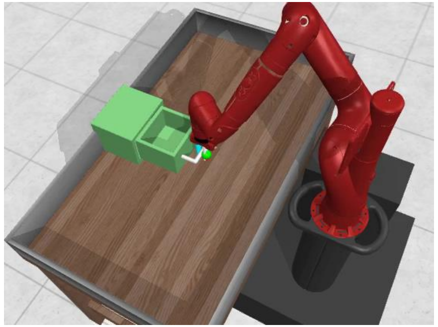
"Find an apple."

Designing a Reward Function

Designing a new reward function for each user and re-training the agent is time-consuming.

Mujoco

MetaWorld



iTHOR

Low ⟷ High

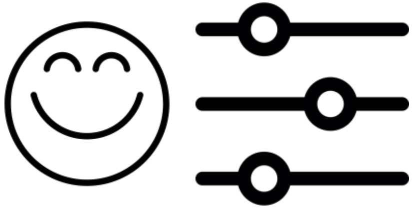
Dynamics DoF, Size of State & Action

Task Difficulty

Objective

1. Can we personalize a policy for human preference over **multiple objectives**?

2. Can we **efficiently estimate** human preference over multiple objectives?



Personalization

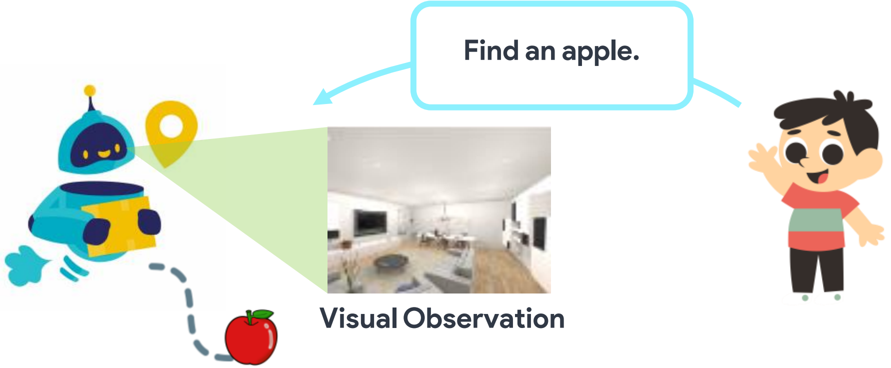
Time & Cost Efficient

Tasks

We introduce two personalized navigation tasks.

1) *Personalized Object-Goal Navigation*

Task: Find the target object while satisfying human's preference over the agent's behavior.

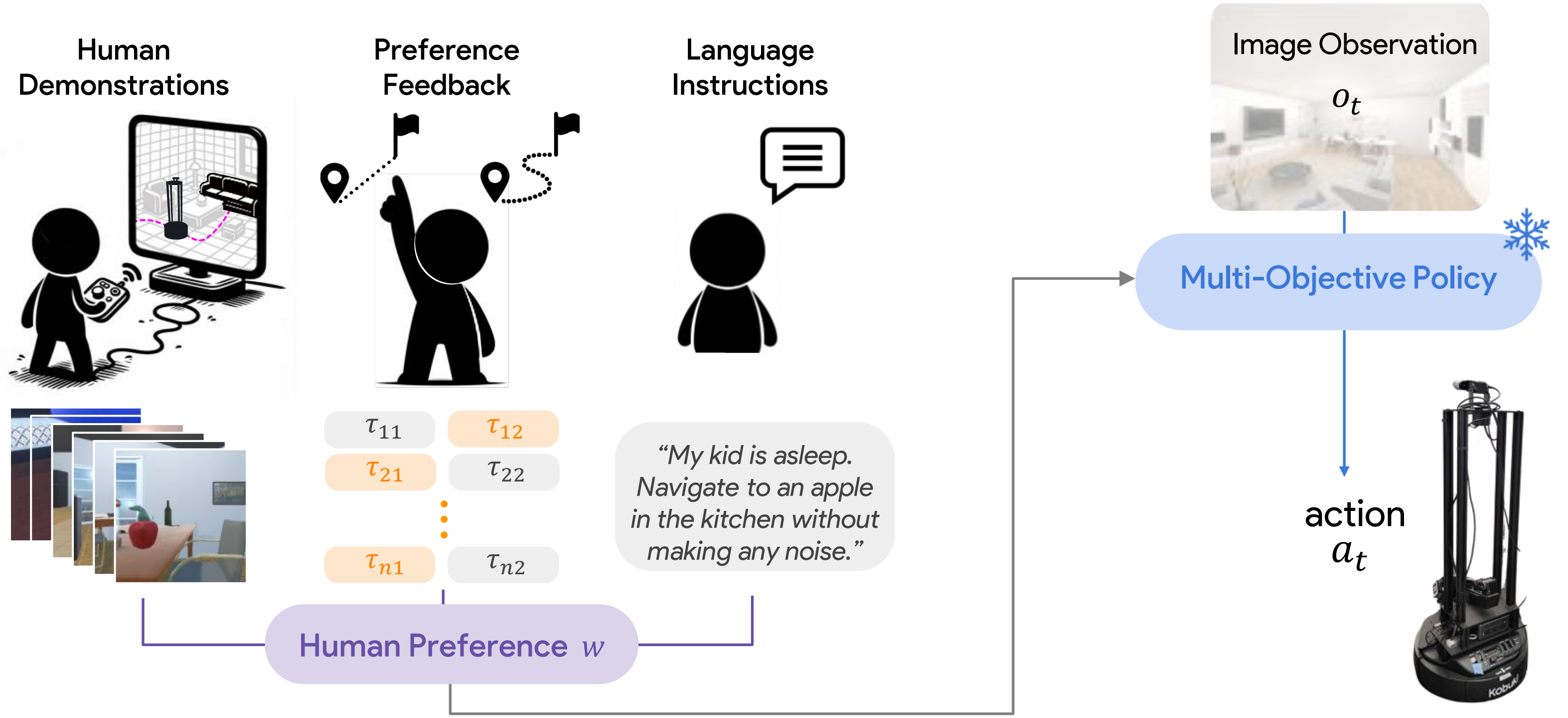


2) *Personalized Flee Navigation*

Task: Run away from the initial location while satisfying human's preference over the agent's behavior.

Promptable Behaviors

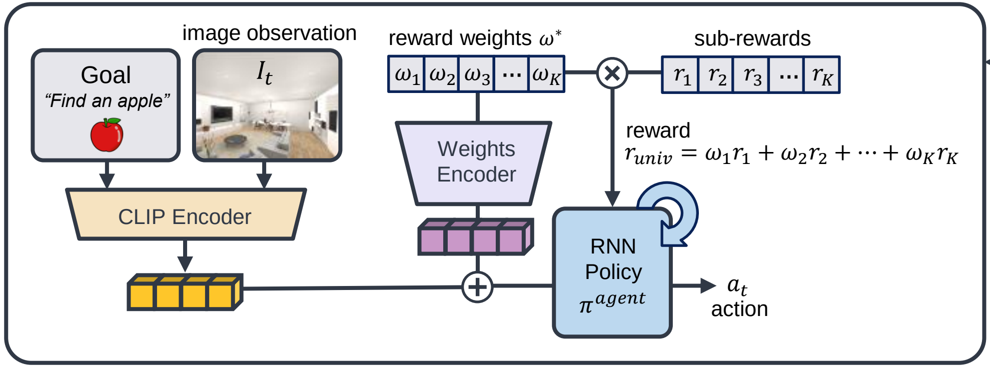
We propose ***Promptable Behaviors***, a novel personalization framework that deals with diverse human preferences **without re-training the agent**.



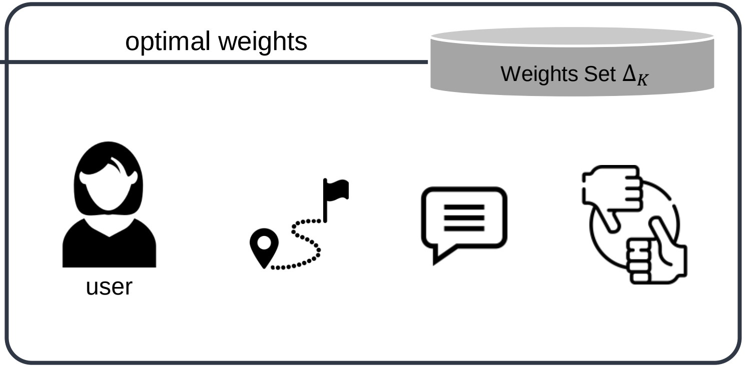
Network Architecture

We take a modular approach:

1) Train a policy conditioned on a reward weight vector across multiple objectives
2) Predict the optimal weights of a human user given human demonstrations/preference feedback/language instructions



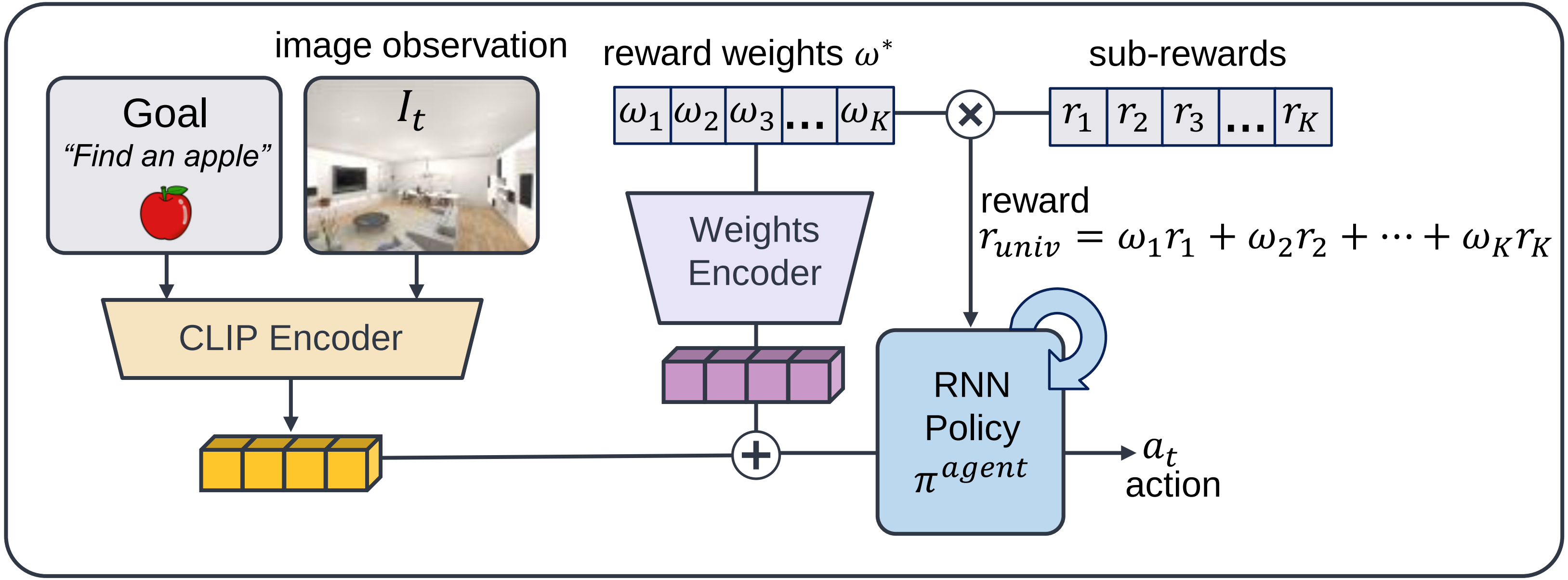
Multi-Objective Policy Training



Predicting Human Preference

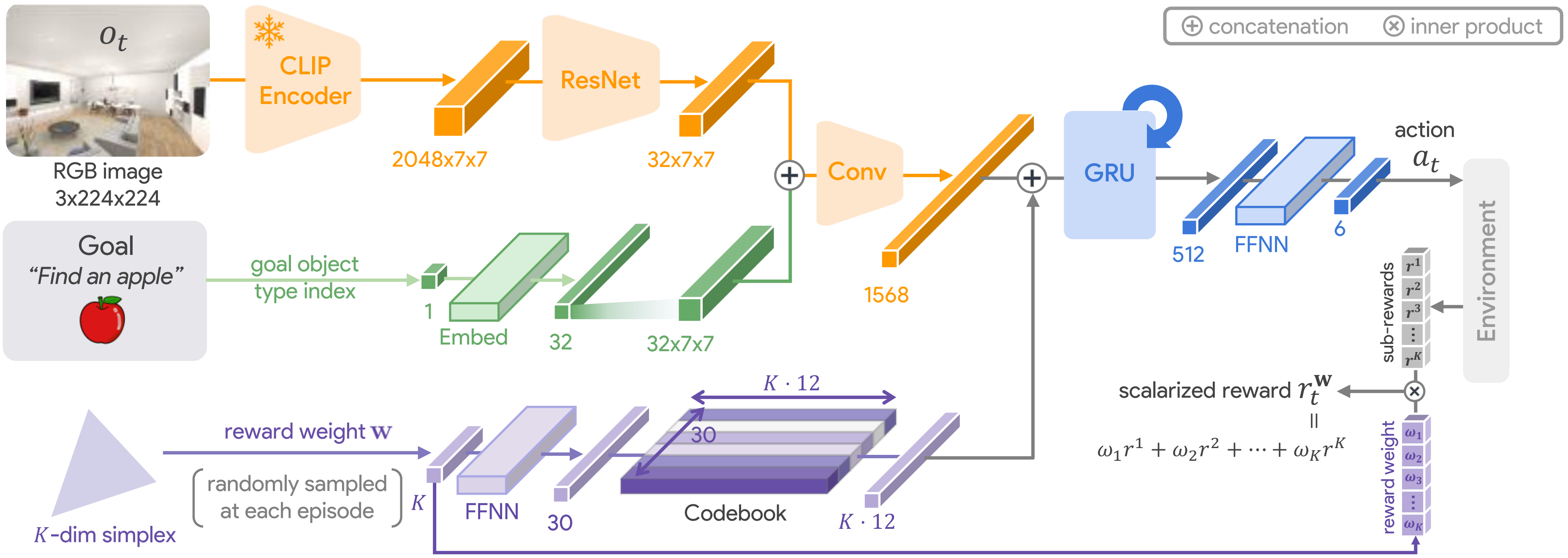
Multi-Objective Policy Training

1) Train a policy conditioned on a reward weight vector across multiple objectives
   We convert multi-objective RL to single-objective RL by reward scalarization



Network Architecture

Specifically, we use a codebook module to encode the reward weight vector.



Experiments

We perform two experiments in two environments (RoboTHOR and ProcTHOR).

Experiment 1) Multi-Objective Policy Training

Preference Objectives:

(ObjectNav) Step Efficiency, Path Efficiency, House Exploration, Safety, and Object Exploration

(FleeNav) Far from Initial, Step Efficiency, House Exploration, and Safety

Experiment 2) Predicting Reward Weights from

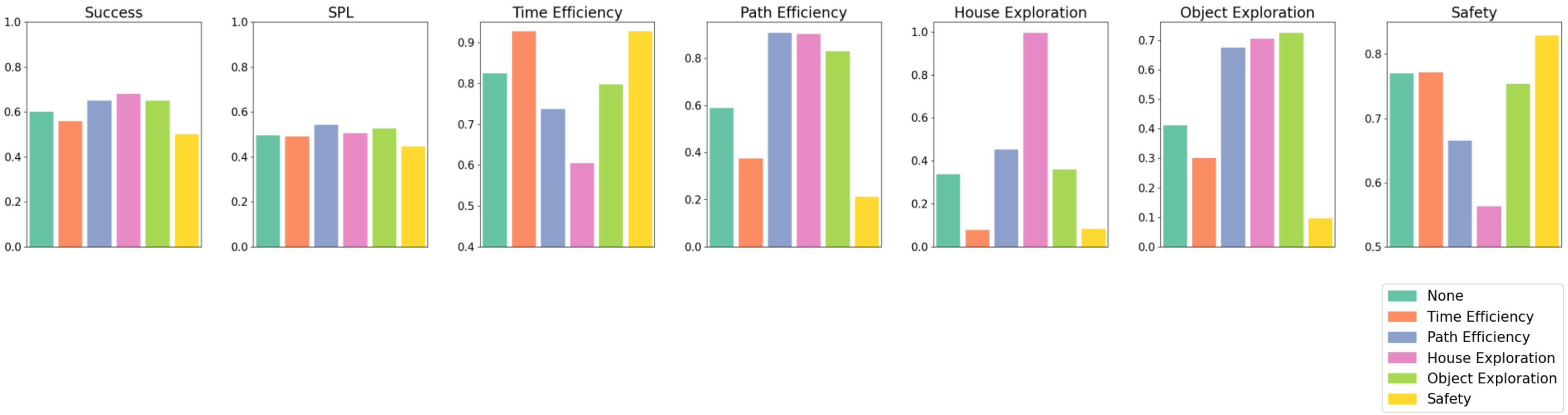
Human Demonstrations / Language Instructions / Trajectory Comparisons

RoboTHOR

ProcTHOR

Experiment 1 – ProcTHOR ObjectNav

Our method achieves high success rates while efficiently optimizing the agent behavior for each objective.



Experiment 1 – ProcTHOR ObjectNav

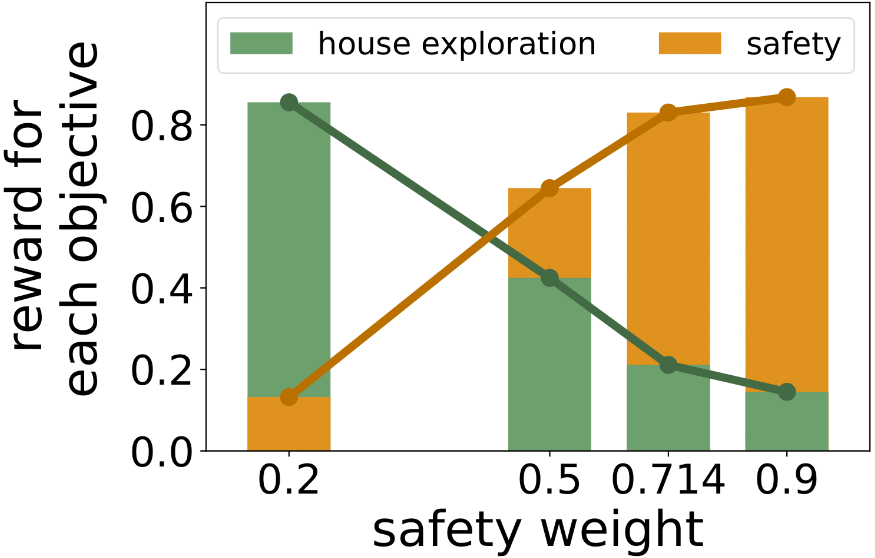
Our method achieves high success rates while efficiently optimizing the agent behavior for each objective.

Method	Multi-Objective		Prioritized Objective	Success ↑	SPL ↑	Distance to Goal ↓	Episode Length ↓	Sub Rewards ↑ Time Efficiency	Path Efficiency	House Exploration	Object Exploration	Safety
EmbCLIP [32]	✗	a	-	0.611	0.455	1.677	105.389	0.767	0.581	0.703	0.731	0.556
Prioritized EmbCLIP	Multi-Policy	b	Time Efficiency	0.560	0.445	2.803	52.060	**0.926**	0.317	0.136	0.247	0.746
		c	Path Efficiency	0.611	0.449	2.038	106.444	0.764	0.515	0.590	0.731	0.693
		d	House Exploration	0.200	0.113	3.921	350.960	0.033	**0.677**	**2.868**	0.161	0.012
		e	Object Exploration	0.611	0.513	2.439	138.389	0.668	0.414	0.703	**0.731**	0.556
		f	Safety	0.480	0.391	3.237	56.620	0.912	0.016	0.130	0.004	**0.834**
Promptable Behaviors (Ours)	Single-Policy	g	-	0.600	0.496	2.526	86.070	0.824	0.589	0.336	0.412	0.770
		h	Time Efficiency	0.560	0.492	2.675	51.760	**0.927**	0.375	0.078	0.301	0.772
		i	Path Efficiency	0.650	**0.543**	2.213	115.350	0.737	**0.907**	0.451	0.674	0.665
		j	House Exploration	**0.680**	0.506	2.253	159.440	0.605	0.902	**0.995**	0.705	0.563
		k	Object Exploration	0.650	0.525	2.198	94.890	0.798	0.829	0.358	**0.725**	0.754
		l	Safety	0.500	0.446	2.875	51.890	0.927	0.211	0.083	0.096	**0.829**

Table 1. **Performance in ProcTHOR ObjectNav.** We evaluate each method in the validation set with six different configurations of objective prioritization: uniform reward weight across all objectives and prioritizing a single objective 4 times as much as other objectives. Sub-rewards for each objective are accumulated during each episode, averaged across episodes, and then normalized using the mean and variance calculated across all methods. Colored cells indicate the highest values in each sub-reward column.

Experiment 1 – ObjectNav

As the safety weight increases, the safety reward increases while the exploration (conflicting objective) reward decreases.



Experiment 1 – ProcTHOR FleeNav

Evaluation results show that the policy is promptable by changing the reward weights.

Method	Multi-Objective		Prioritized Objective	Success ↑	PLOPL ↑	Distance to Furthest ↓	Episode Length ↓	Sub Rewards ↑ Time Efficiency	House Exploration	Safety
Prioritized EmbCLIP	Multi-Policy	a	Time Efficiency	0.691	0.810	7.360	57.090	**0.875**	0.420	0.138
		b	House Exploration	**0.759**	0.872	6.704	58.330	0.839	**0.835**	0.215
		c	Safety	0.723	0.856	7.391	57.640	0.859	0.676	**0.487**
Promptable Behaviors (Ours)	Single-Policy	d	-	0.700	0.805	7.013	69.020	0.531	0.365	0.522
		e	Time Efficiency	0.728	0.832	6.592	66.490	**0.604**	0.434	0.563
		f	House Exploration	0.737	0.861	6.317	71.500	0.460	**0.813**	0.089
		g	Safety	0.711	0.814	6.735	67.830	0.566	0.227	**0.776**

Table 2. **Performance in ProcTHOR FleeNav.** We evaluate each method in the validation set with five different configurations of objective prioritization: uniform reward weight across all objectives and prioritizing a single objective 3 times as much as other objectives. The displayed sub-reward values are normalized for each objective following Table 1.

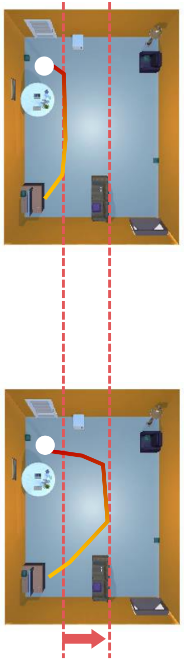
Experiment 1 – Visualization

The agent shows different trajectories based on object prioritization.

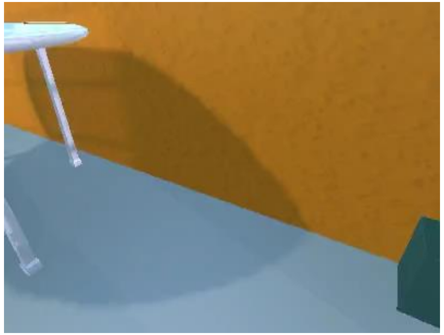
Time Efficiency vs House Exploration

Time Efficiency

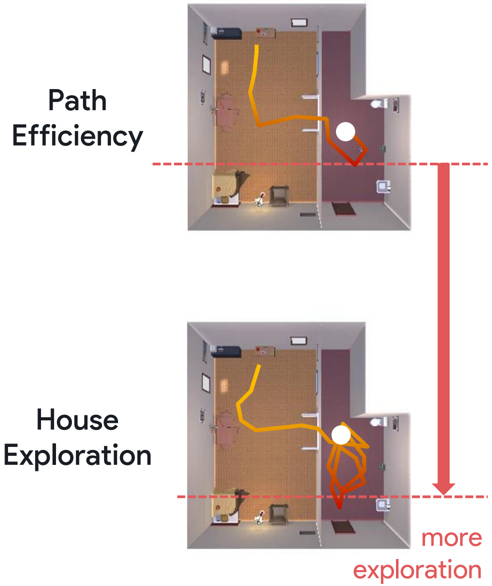
House Exploration

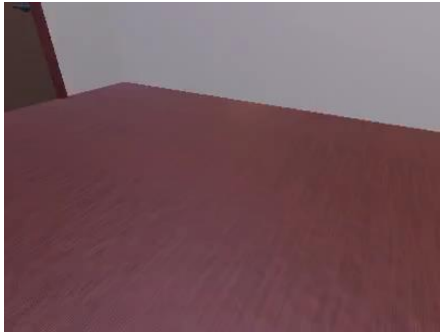


more exploration



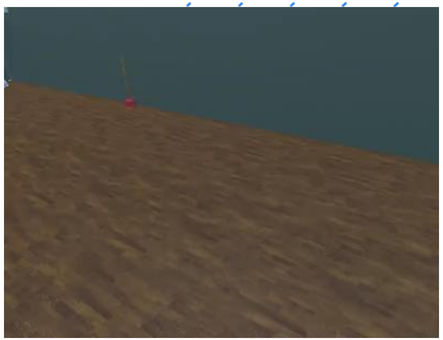
Path Efficiency vs House Exploration





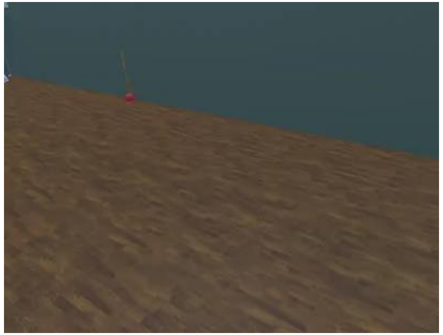
Time Efficiency vs Path Efficiency

Time Efficiency



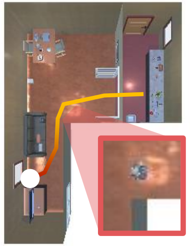
Path Efficiency

more rotation actions

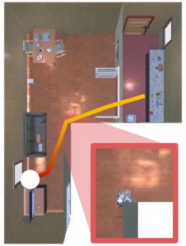


Time Efficiency vs Path Efficiency

Time Efficiency

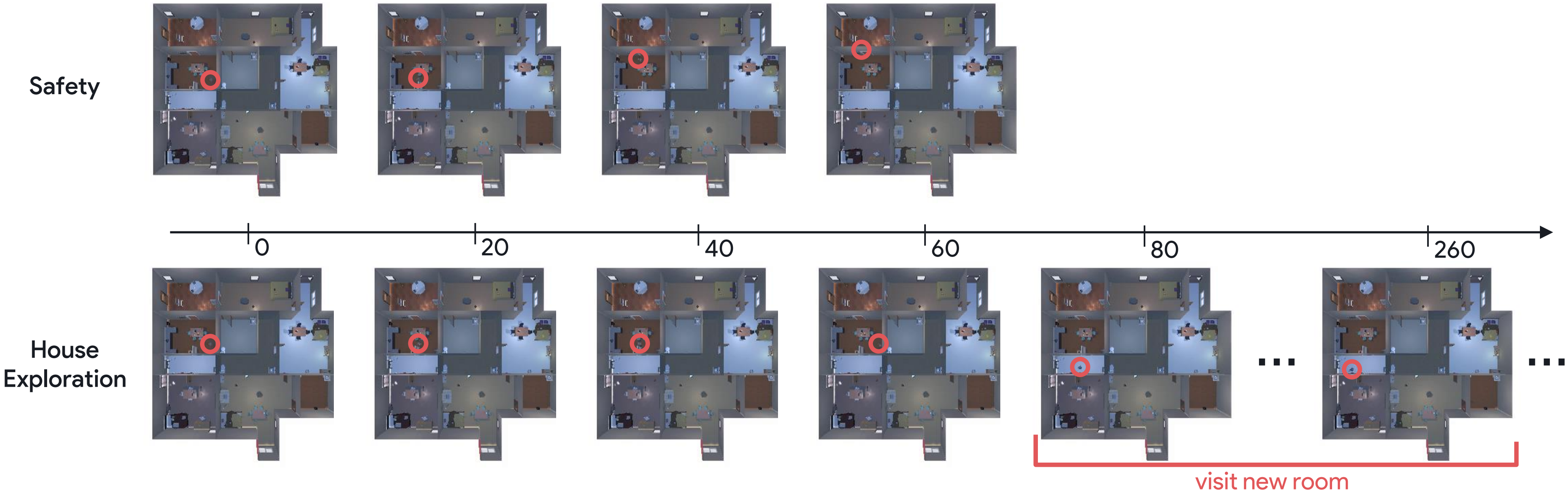


Path Efficiency

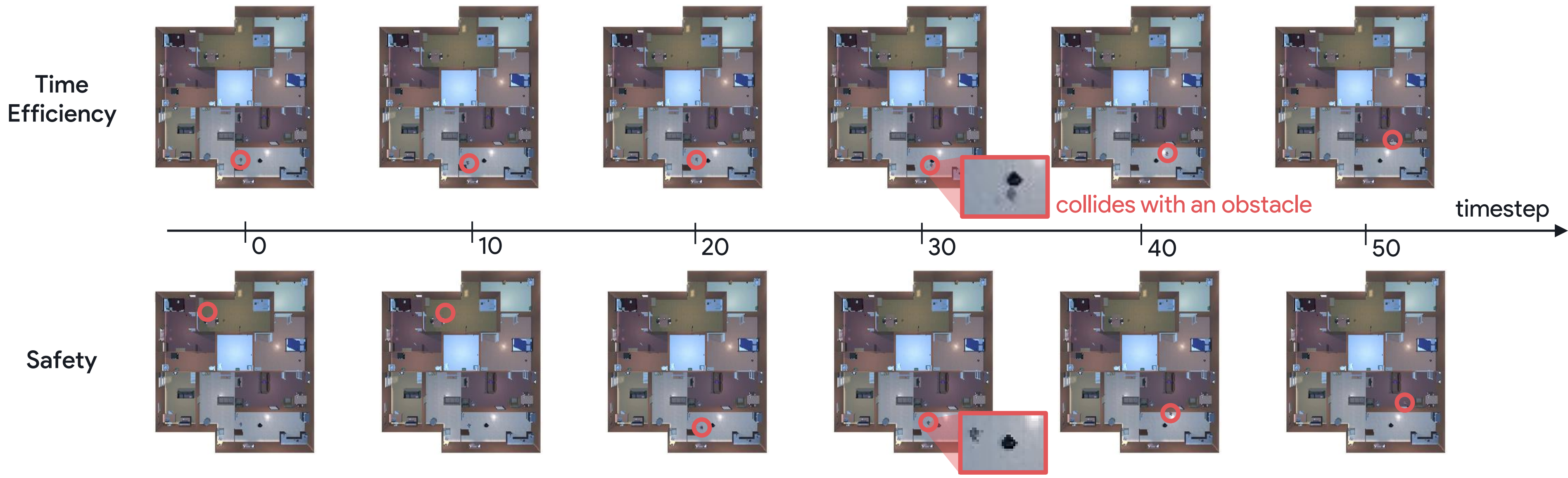


moves closer to the wall

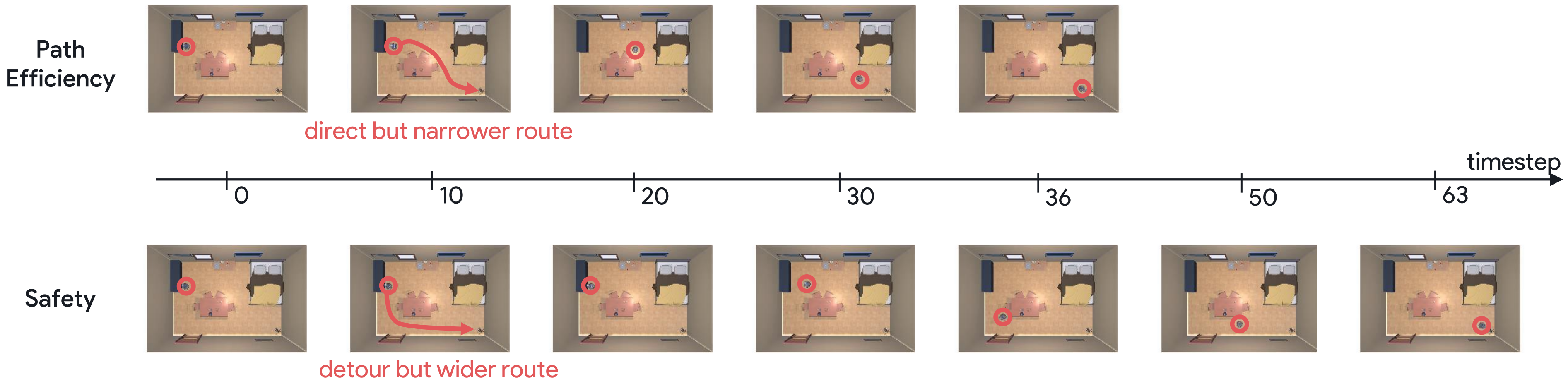
House Exploration vs Safety



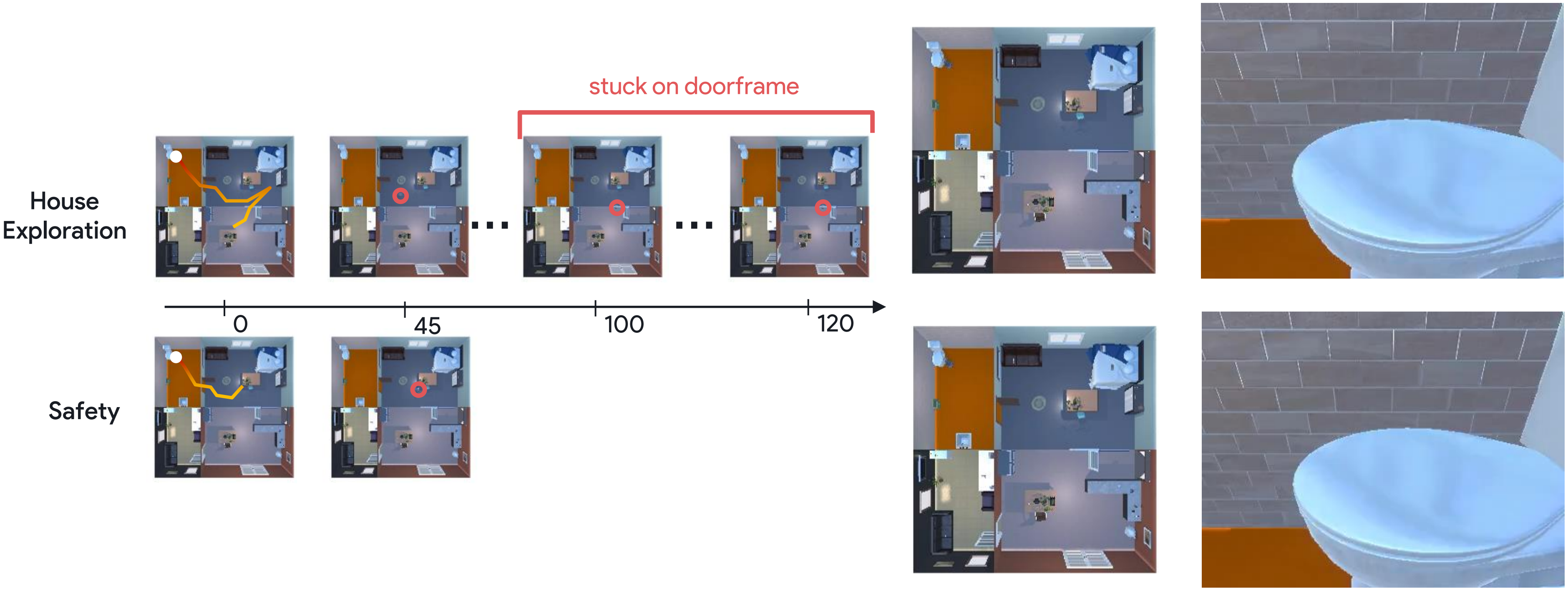
Time Efficiency vs Safety



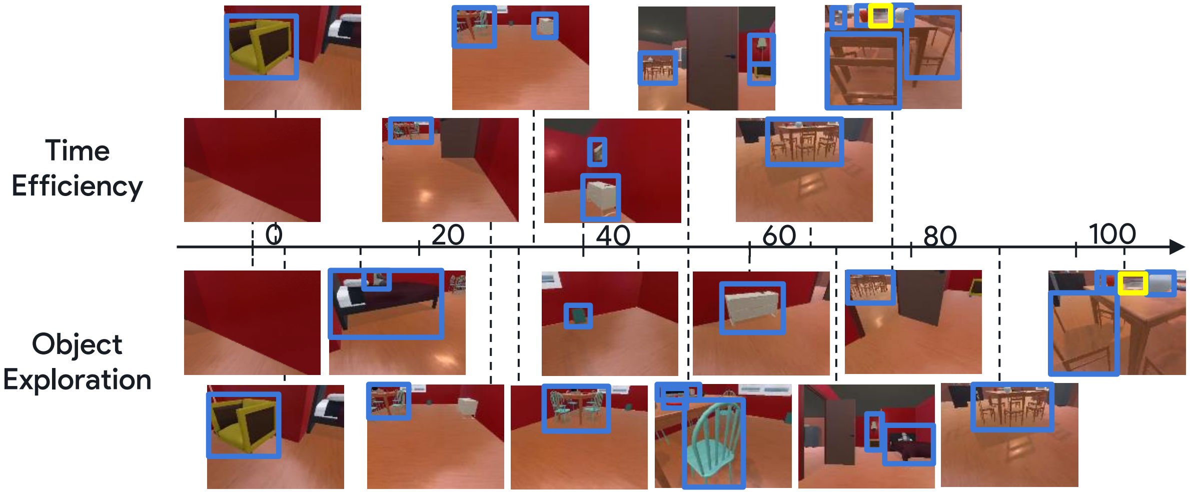
Path Efficiency vs Safety



House Exploration vs Safety



Time Efficiency vs Object Exploration



Human Preferences to Reward Weights

We provide a variety of options for users to provide their preferences to the agent. Specifically, we introduce three distinct methods of reward weight prediction.

Human User

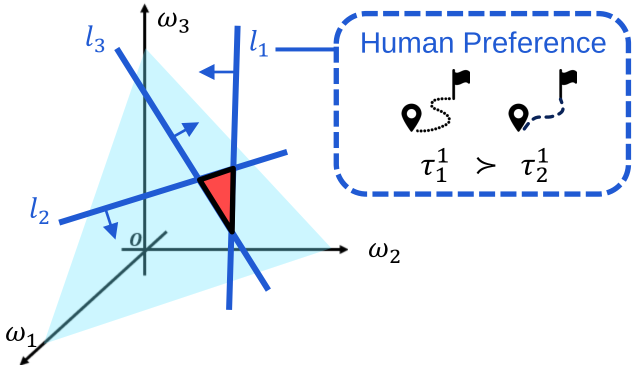
Human Demonstrations

Language Instructions

"Move safely and don't collide with objects or walls."

"Explore every room and try to search as much area as you can."

Human Feedback



Human Demonstrations to Reward Weights

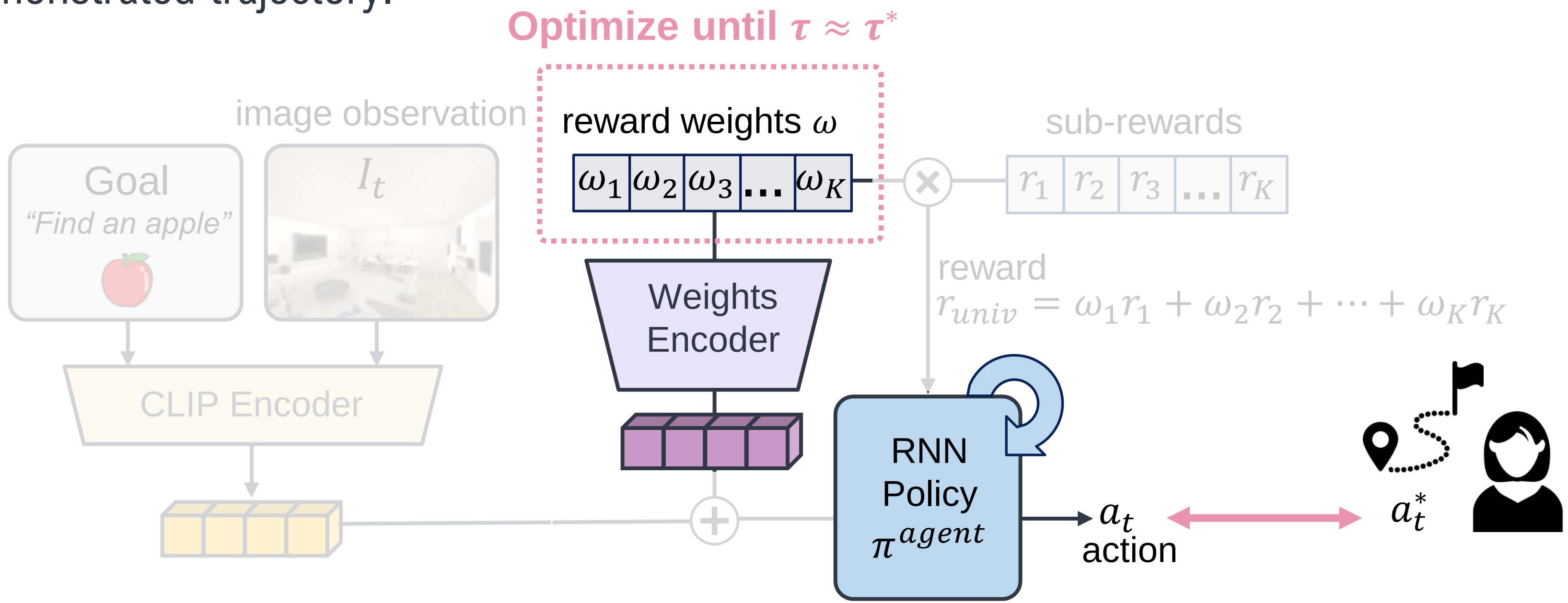
Predict the optimal weights of a human user given a single demonstration.

$$\omega^* = \arg\min_{\omega \in W} \sum loss(\pi(s_t, \omega), a_t^*)$$

Weights Set $W \subset \mathbb{R}^K$

Human Demonstrations to Reward Weights

We optimize the reward weight vector until agent trajectory gets close enough to the demonstrated trajectory.



Language Instructions to Reward Weights

- Use LLM to generate data and predict reward weights
- Chain-of-Thought (CoT) reasoning / In-Context Learning (ICL)

"Prioritize examining objects, even if it takes longer."

or

"After rearranging the house, the user does not remember where certain objects were placed. The user wants to find a specific object, while also inspecting other areas to confirm the new arrangement."

```
Task description
Definition of objectives
Infer reward weights for {instruction}
```

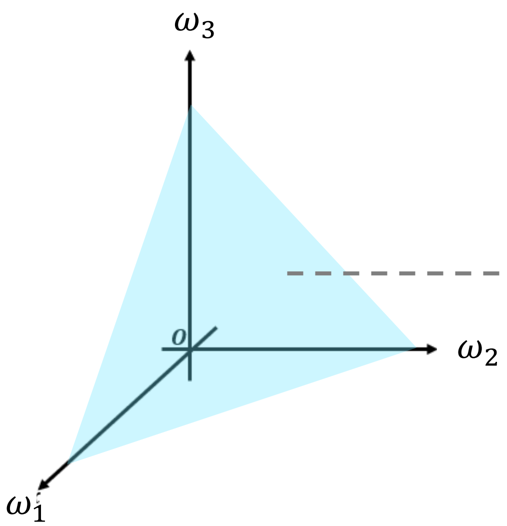
time efficiency: 0.1, path efficiency: 0.1,
house exploration: 0.2, object exploration: 0.5, safety: 0.1

Human Preferences to Reward Weights

Pairwise Comparison

- Use preference data among N trajectory pairs and optimize the reward weights

$\omega_1 + \omega_2 + \omega_3 = 1$ simplex



$$\mathcal{S} = \{(\tau_1^i, \tau_2^i) | \forall i \in \{1, 2, ..., n\}, (\tau_1^i, \tau_2^i) s.t. \ \tau_1 \succ \tau_2\}$$

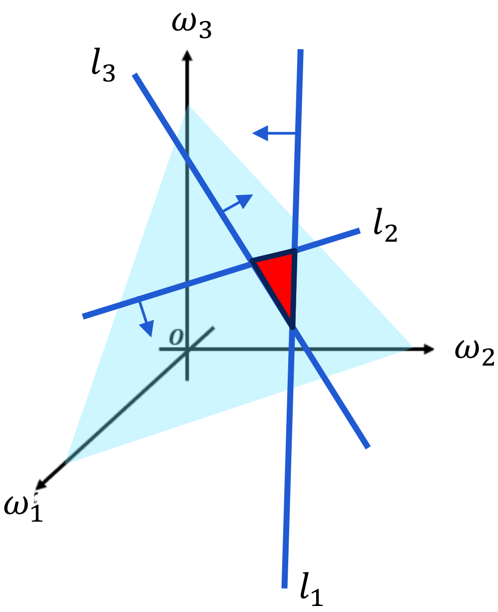
$$\boxed{\begin{aligned} \omega^* &= \arg\max_{\omega \in \mathcal{W}} \mathop{\mathbb{E}}_{(\tau_1, \tau_2) \in \mathcal{S}} [P(\tau_1 \succ \tau_2)] \qquad (10) \\ &= \arg\max_{\omega \in \mathcal{W}} \mathop{\mathbb{E}}_{(\tau_1, \tau_2) \in \mathcal{S}} \left[\log \left(\frac{\exp r(\tau_1)}{\exp(r(\tau_1) + \exp(r(\tau_2)))} \right) \right] \end{aligned}}$$

Maximize the log-likelihood of preferences

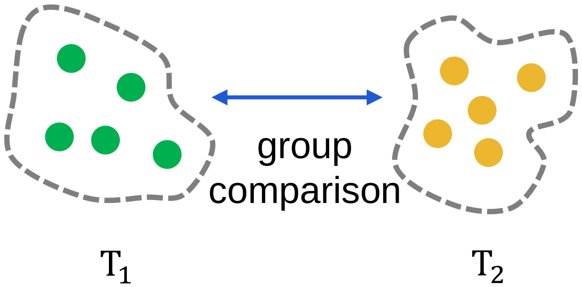
Human Preferences to Reward Weights

We also propose group trajectory comparison, which significantly reduces the labeling effort by allowing users to compare groups of trajectories.

$\omega_1 + \omega_2 + \omega_3 = 1$ simplex



$$\tau_1 \succ \tau_2 := r(\tau_1) > r(\tau_2) = \omega^\intercal \vec{r}(\tau_1) > \omega^\intercal \vec{r}(\tau_2) \tag{8}$$



Reward Weight Prediction Results

Utilizing preference feedback is the most accurate, while using language instructions is the simplest method.

Weight Prediction Methods: Input	Model	N	Sim ↑	GGI
Human Demonstrations	-	1	0.707	0.347
Preference Feedback	Pairwise Comparison (M=1)	20	0.356	0.800
		50	0.358	0.800
		500	0.897	0.800
	Group Comparison (M=2)	5	0.689	0.626
		10	0.793	0.618
		25	**0.935**	0.657
	Group Comparison (M=5)	2	0.722	0.634
		4	0.682	0.762
		10	0.862	0.641
Language Instructions	ChatGPT	1	0.530	0.388
	w/ ICL	1	0.529	0.379
	w/ CoT	1	0.614	0.391
	w/ ICL + CoT	1	0.482	0.347

Table 3. Comparison of Three Weight Prediction Methods

Weight Prediction Methods: Input	Model	N	Win Rate ↑
Human Demo.	-	1	0.556
Preference Feedback	Pairwise Comparison (M=1)	50	0.552
	Group Comparison (M=2)	25	**0.650**
	Group Comparison (M=5)	10	0.588
Language Instruction	ChatGPT w/ CoT	1	0.600

Table 4. Human Evaluation on Scenario-Trajectory Matching

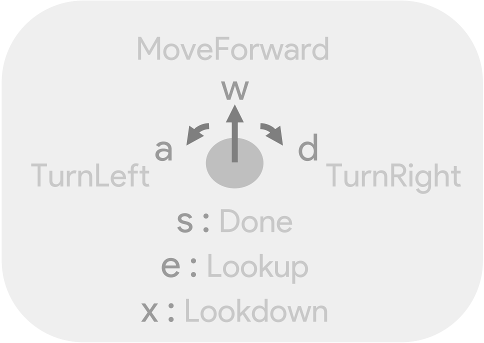
Full-Framework Demo

Demo – Human Demonstration

Demonstrate a trajectory that fits your preference.

Max time horizon: 500

Scenario: "I just moved in and want to find which furniture or object is located while inspecting the layout of the house as a video."



Predicting Reward Weight from Human Demonstration ...

Finished!

[Predicted Reward Weight]

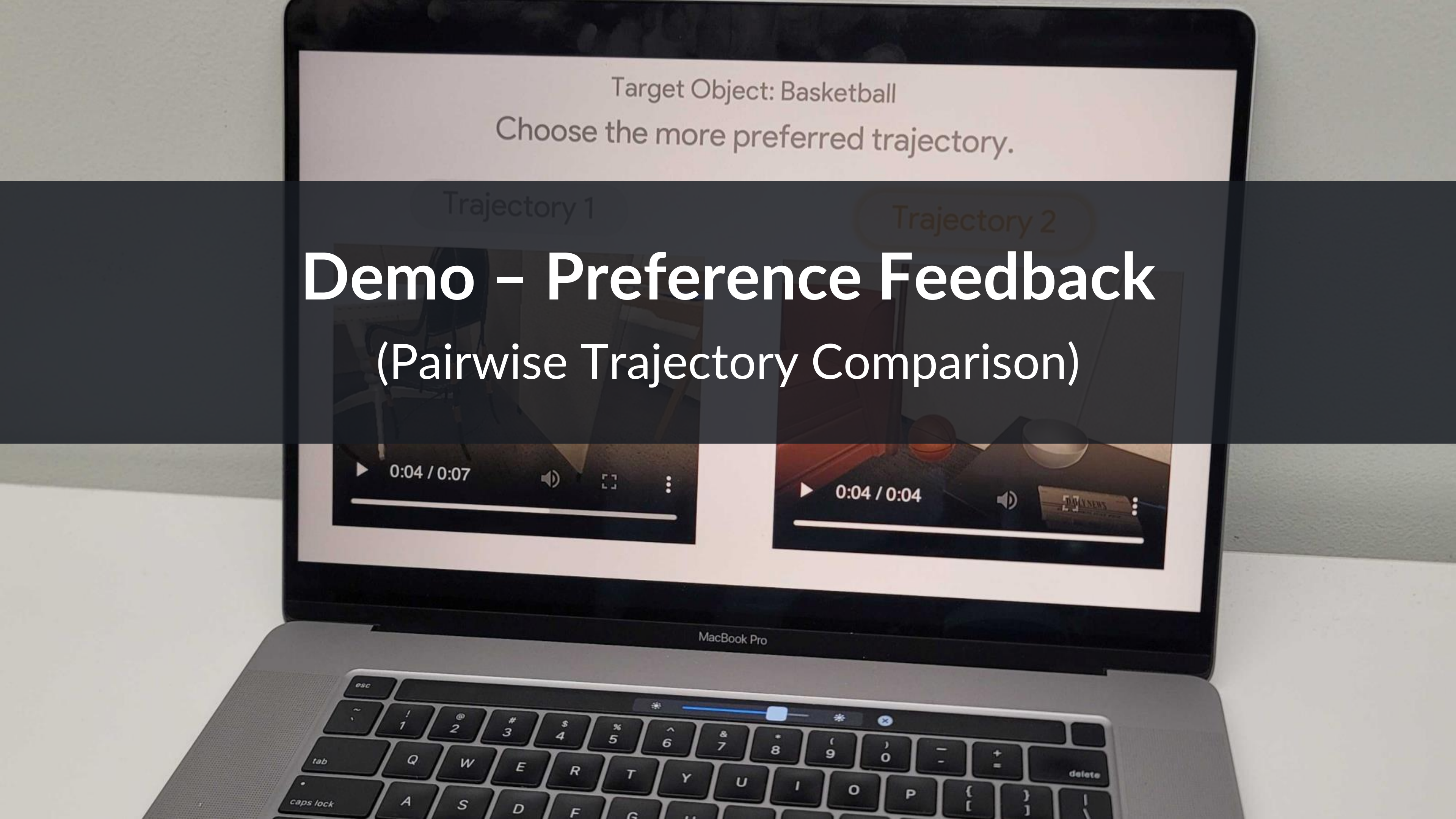
time effiency : path efficiency : **house exploration** : object exploration : safety
= 0.087 : 0.210 : **0.463** : 0.093 : 0.147

RGB Image Observation

Top-Down View

Demo – Preference Feedback

(Pairwise Trajectory Comparison)



Scenario: "I want to check an appliance in the house while I'm away, but the robot has a low battery. I don't want the robot to waste its battery while looking into unnecessary regions."

Choose the more preferred trajectory.

Trajectory 1

Trajectory 2

[Episode 1] Target Object: Basketball

Predicting Reward Weight from Preference Feedback on Pairwise Trajectory Comparisons ...

Finished!

[Predicted Reward Weight]

time effiency : path efficiency : house exploration : object exploration : safety

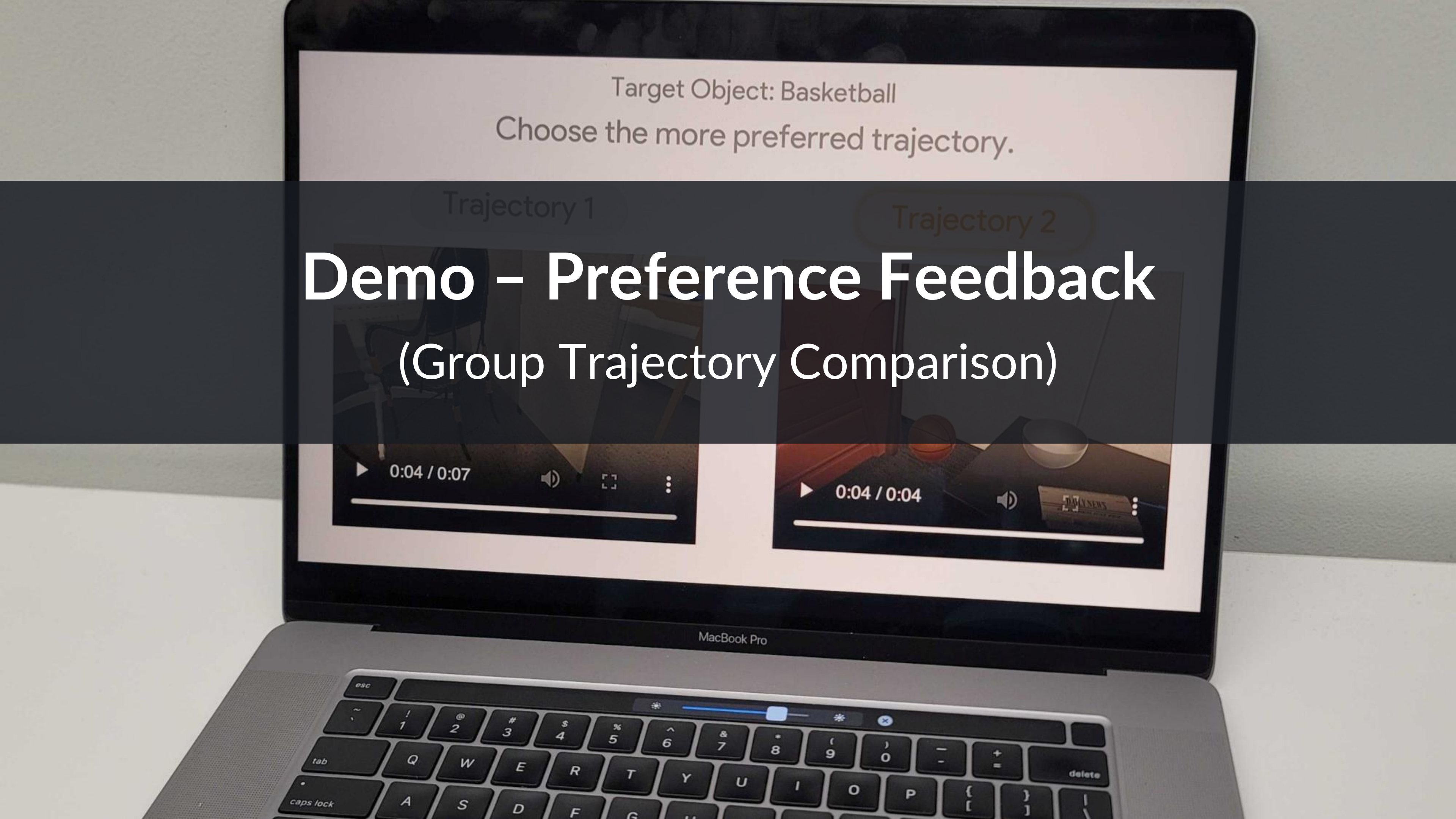
= 0.0 : 0.682 : 0.127 : 0.057 : 0.134

RGB Image Observation

Top-Down View

Demo – Preference Feedback

(Group Trajectory Comparison)



Choose the more preferred trajectory group.

Group 1

Safety weight ≥ 0.5

Group 2

House Exploration weight ≥ 0.5

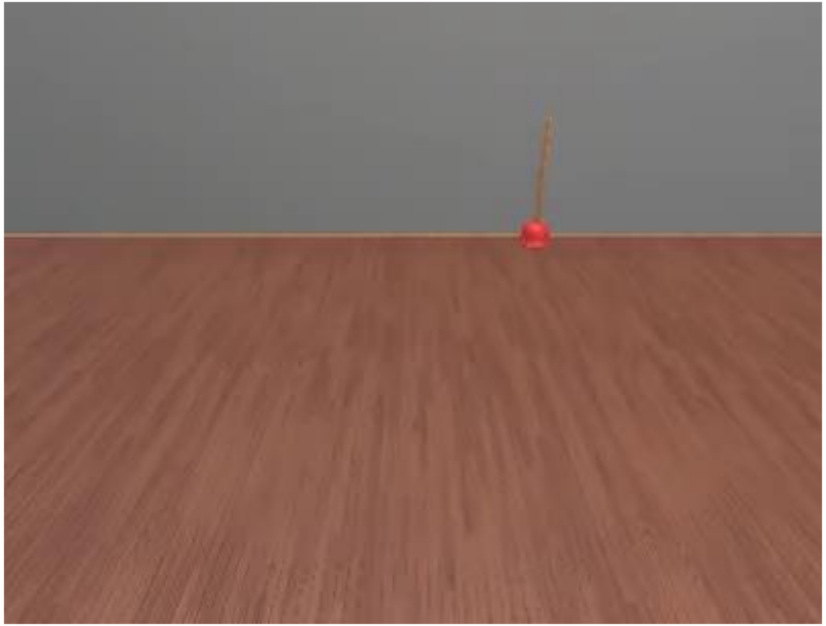
Predicting Reward Weight from Preference Feedback
on Group Trajectory Comparisons ...

Finished!

[Predicted Reward Weight]

time effiency : path efficiency : house exploration : object exploration : safety

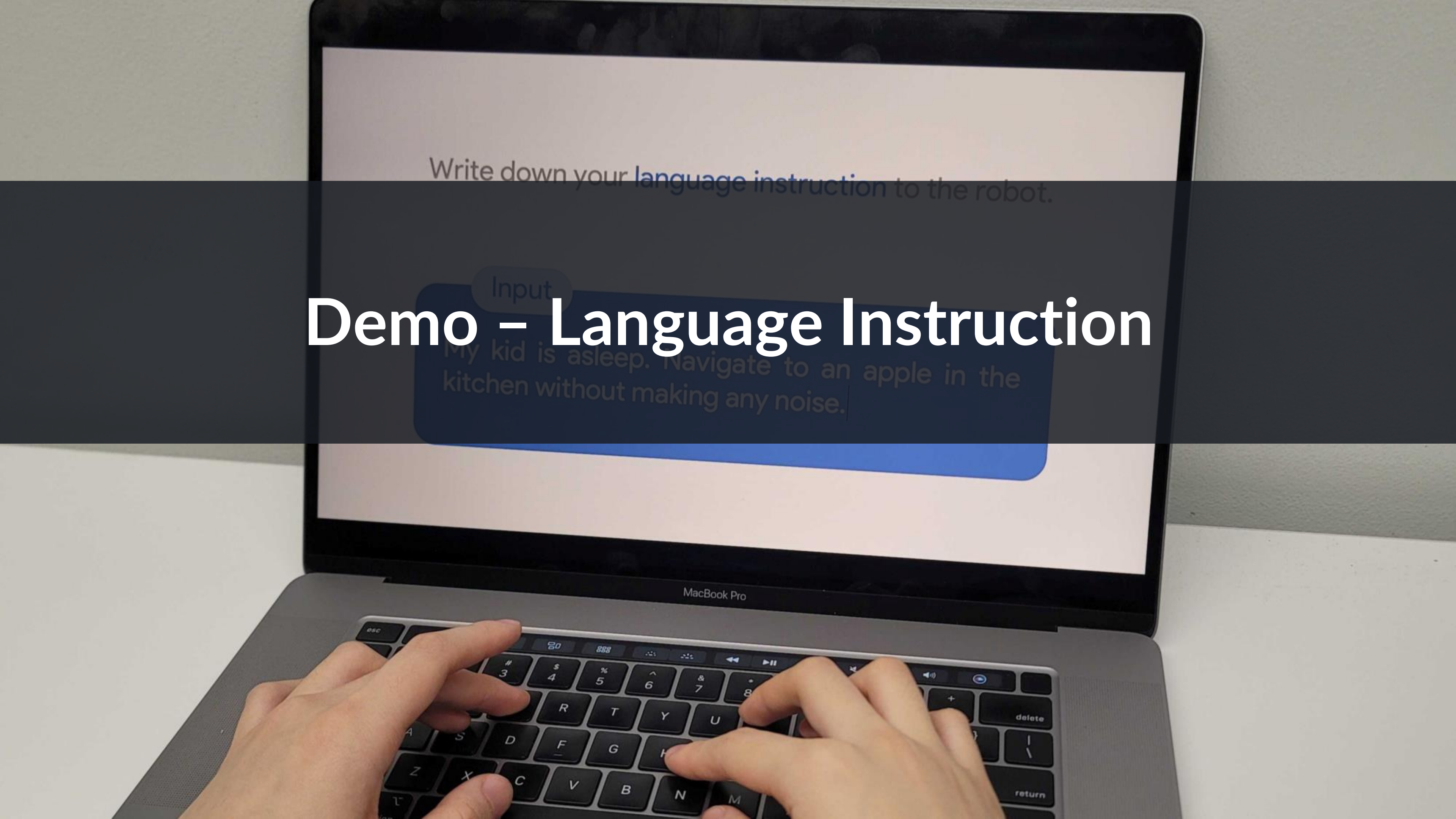
= 0.029 : 0.813 : 0.036 : 0.051 : 0.071



RGB Image Observation

Top-Down View

Demo – Language Instruction



Write down your language instruction to the robot.

Input

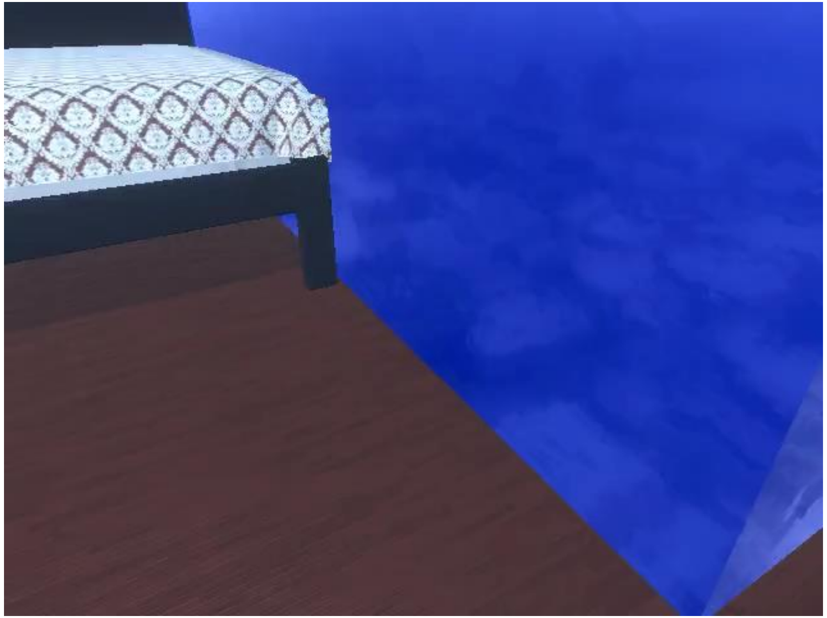
Predicting Reward Weight from Language Instruction ...

Finished!

[Predicted Reward Weight]

time effiency : path efficiency : house exploration : object exploration : safety

= 0.1 : 0.1 : 0.1 : 0.1 : 0.6



RGB Image Observation

Top-Down View

Contributions

1) A novel framework for **personalized learning** that enables robots to align with diverse human preferences in complex embodied AI tasks **without any policy fine-tuning**.

2) Three methods for **inferring human preferences** using human demonstrations, preference feedback on trajectory comparisons, and language instructions, each offering unique advantages.

3) Demonstrations in two long-horizon personalized navigation tasks shows the effectiveness of our approach in prompting agent behaviors to satisfy human preferences.

Thank you.

Code, Paper, and Visualizations available at:

Project Website