Under Review

Latent Preference Reasoning for Multi-Session Personalized Tool-Calling

Yejin Yoon * Minseo Kim * Taeuk Kim †

Hanyang University, Seoul, Republic of Korea

Presented by Yejin Yoon

HYU 한양대학교 HANYANG UNIVERSITY

Memory?

External, symbolic, non-parametric memory

Unlike internal state-based memory (e.g., LSTM hidden states) or parametric memory stored in model weights, we focus on **external**, **non-parametric memory** that supports governed preference reasoning at inference time.

WHY — *Why do we need memory?*

- Retain information beyond limited context windows
- Generalize from individual interactions to stable patterns
- Stabilize model behavior across turns or sessions
- Reduce repeated reasoning and inference cost

WHEN — *When is memory used?*

- Training-time: absorbed into model parameters
- Inference-time: supports test-time reasoning
- Within-session: short-term coherence
- Across sessions: long-term personalization

WHAT — *What is stored?*

- Raw text (utterances, documents)
- Structured attributes (profiles, statements)
- Representations (embeddings, hidden states), …

HOW — *How is memory constructed and updated?*

- Appending and retrieving past information
- Summarization or compression of history
- Overwriting or updating states

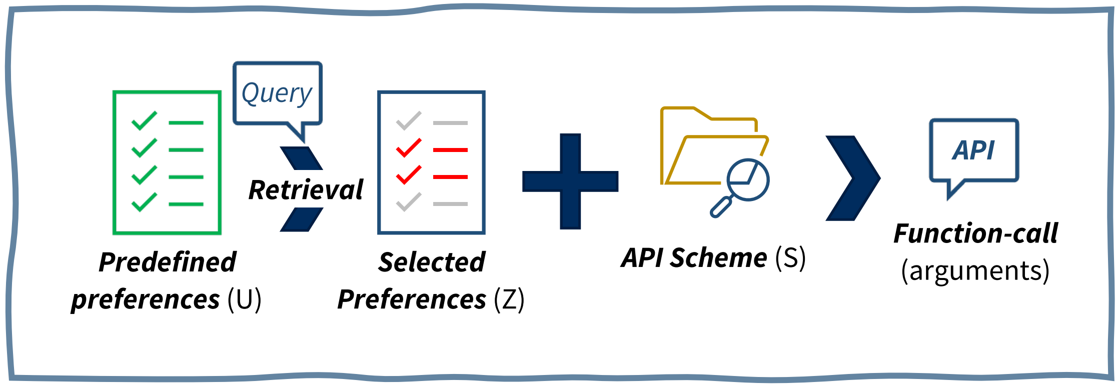
Key Challenge: What does it mean to call something "**Memory**" in modern NLP systems?

Motivation

Personalized tool-calling : what's missing?

Tool-calling agents increasingly rely on user preferences to resolve **underspecified** arguments

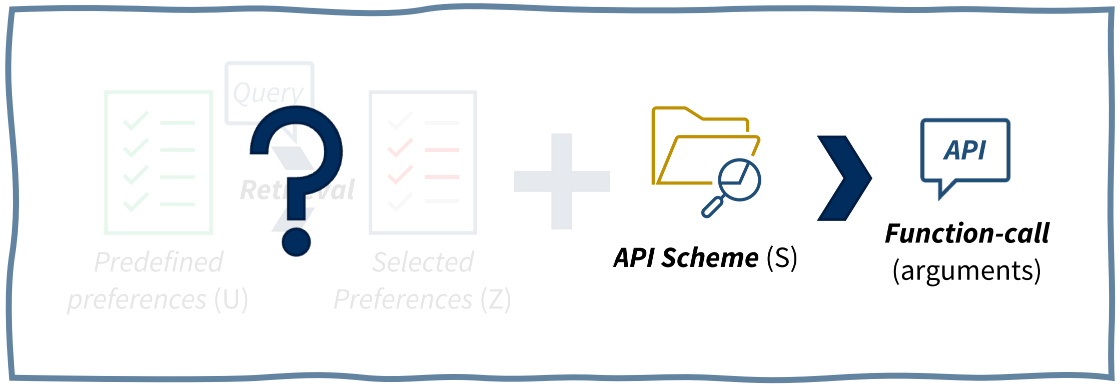
Tool-calling agents:



Existing benchmarks assume preferences are:

- **explicitly** stated, or
- provided as **static** user profiles

In reality:



In real interactions, preferences are:

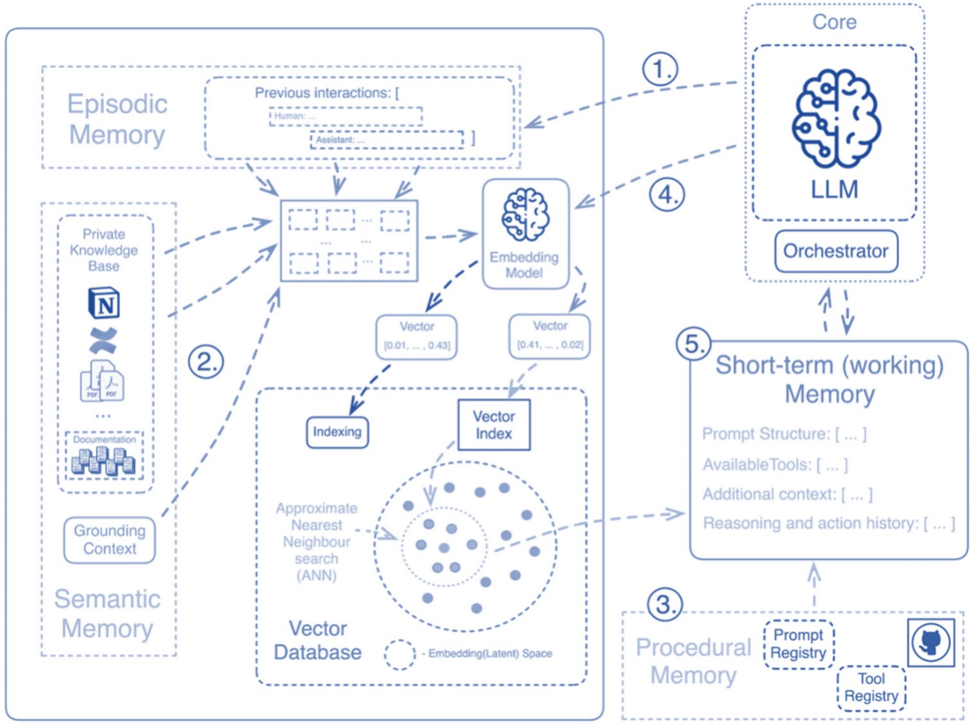
- **rarely** stated upfront
- revealed **implicitly** through repeated user behavior

🤔 A gap between benchmark assumptions and real-world usage

Motivation

Limitation of Memory-based approaches : 🔎 retrieval is not enough

Recent agents incorporating **memory** or **retrieval** to support personalization



Can **preferences** be considered **surface facts**?

- **Retrieval-based agents' memory** captures :
  - *surface-level utterances*
  - *Isolated factual preferences*

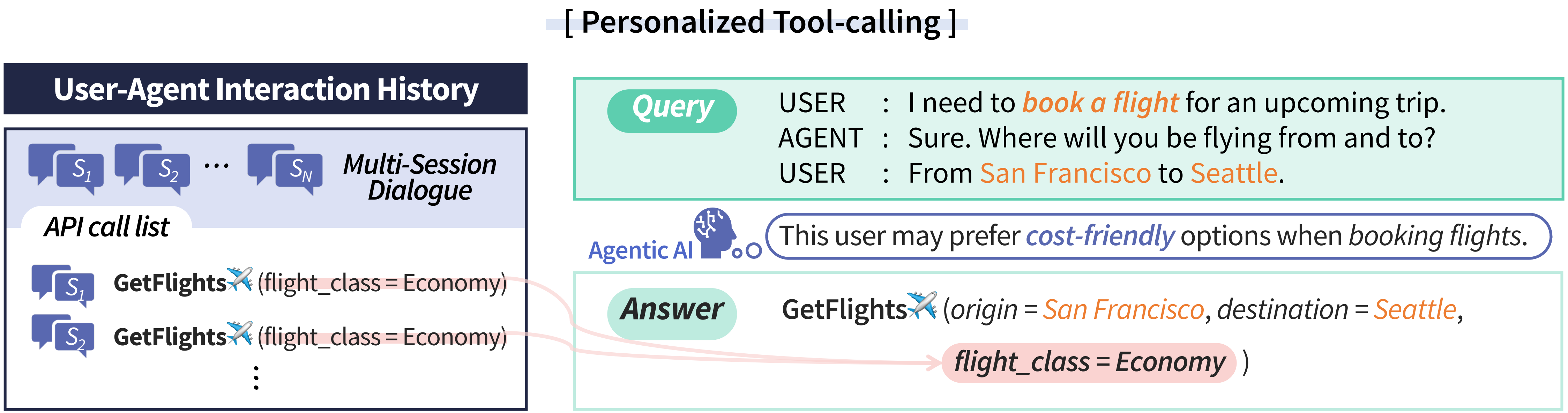
- **Preferences** are often :
  - *Abstract* (e.g., cost-sensitive, brand-loyal)
  - *Indirect* (expressed through choices, not statements)
  - *Cross-domain* (applicable across tools and tasks)

⁉️ Retrieving past utterances does not guarantee **correct preference usage**

Problem Definition

⛳ Our Goal : Multi-session personalized tool-calling

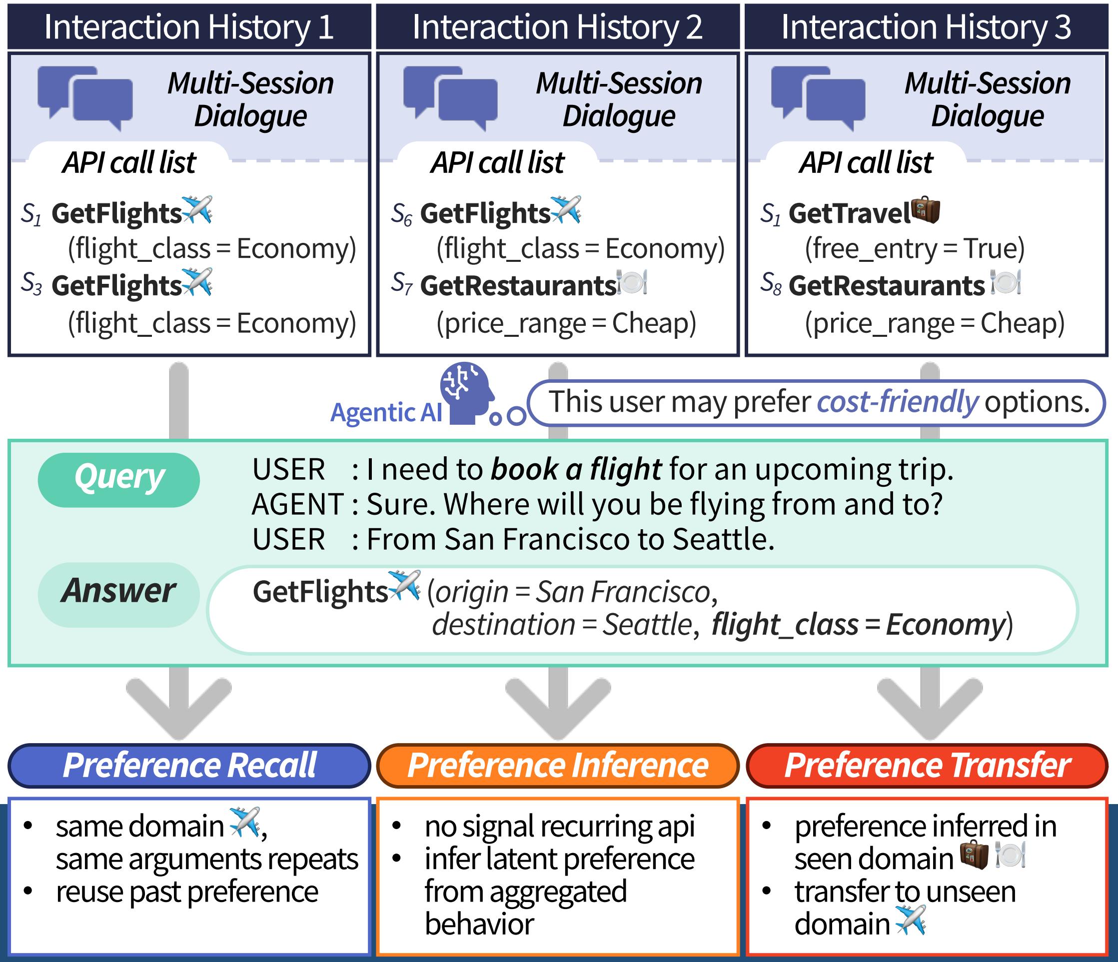
- **Input** : current query + interaction history spanning multiple sessions
- **Output :** correct API call with missing or underspecified arguments



Key Challenge: correct arguments depend on preferences that are never explicitly stated

Preference Reasoning Types

Preference Reasoning Design : 3 reasoning types require qualitatively different model behavior



* Latent Preference Reasoning

- *Latent Preferences*: Action-level constraints that emerge from user behavior accumulated across sessions.
- *Preference Reasoning*: The process of determining, at tool-call time, which of the available options is most consistent with the user's past behavior.

* Research Questions

- Can an agent, for each individual user:
  1) infer unstated (latent) preferences from interaction history?
  2) abstract beyond individual actions to form generalizable preference representations?
  3) apply these preferences to tool selection even in unseen domains?

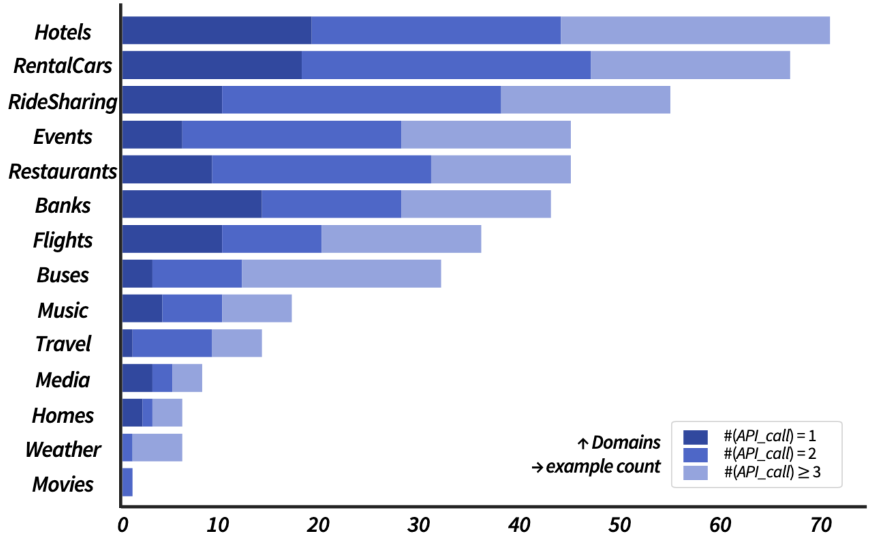
A **single history** can support multiple types depending on the **query**

Benchmark Construction Overview (MPT)

What we construct : a benchmark for evaluating latent preference reasoning

- One interaction history paired with multiple queries.
- Each query is designed to probe a specific preference reasoning type.

[Domain Distribution]



[Group Preference API]

Group	Preference	API(arguments)
Budget	low_cost	GetRestaurants(price_range = cheap) GetRentalCars(car_type = Compact) GetHotels(average_star = 1,2) GetRideSharing(shared_ride = True) GetTravel(free_entry = True) GetFlights(flight_class = Economy)
	high_cost	GetRestaurants(price_range = pricey) GetRentalCars(car_type = Full-size) GetHotels(average_star = 4,5)
Travel	solo	GetBuses(group_size = 1) GetFlights(passengers = 1) GetRideSharing(number_of_seats = 1) GetEvents(number_of_tickets = 1) GetRestaurants(number_of_seats = 1)

[Instance Sample]


```
{
  "example_id": "...",
  "sessions": [ ... ],
  "api_calls_pref": [
    {
      "group_preference": "budget_conscious",
      "value_group": "high_cost",
      "count": 6,
      "evidence": [
        {
          "domain": "GetHotels",
          "slot": "average_star",
          "values": [
            {"value": "4", "count": 4},
            {"value": "5", "count": 2}
          ]
        }
      ]
    },
    {
      "group_preference": "travel",
      "value_group": "solo_usage",
      "count": 3,
      "evidence": [
        {"domain": "GetFlights", "slot": "passengers",
        "value": "1"},
        {"domain": "GetEvents", "slot": "number_of_tickets",
        "value": "1"}
      ]
    }
  ]
}
```


Agents must (1) decide whether a preference is relevant to the current query
(2) infer preferences from indirect evidence (3) generalize preferences across domains

Experimental Setup

Modeling approaches for handling personalized tool-calling

1. **Vanilla LLM** (GPT-5, Gemini-3, Qwen-3, R1-Distill) – relies entirely on *long-context reasoning*
2. **RAG** (FnCTOD) – *retrieves relevant* past utterances ; lacks abstraction and generalization
3. **Memry Agent** (Mem0) – store *factual preferences* ; limited in handling indirect or cross-domain signals
4. **PRefine** (ours) – *reasons over* evidences ; maintains supported preference abstraction

---

Preference Reasoning — P-EM

Preference Exact Match – measures correctness of inferred latent preferences

- Evaluates whether the model correctly identifies the user's latent preference
- Requires exact matching between the inferred preference and the ground-truth abstraction
- Directly measures preference reasoning accuracy, independent of tool execution

Tool Execution — EA-F1

Execution Argument F1 – measure correctness of tool-call arguments

- Evaluates the correctness of tool-call arguments generated by the model
- Measures whether inferred preferences are correctly applied at execution time
- Reflects downstream task utility under underspecified queries

Evaluation focuses on test-time preference reasoning; no additional fine-tuning or retraining

Main Results

Baseline Observation : preference reasoning with full dialogue history

Vanilla LLM

- Provided with the full accumulated dialogue history, baseline LLMs show:
  - relatively strong performance on **Preference Recall**
  - but substantial performance degradation on harder reasoning types
- Performance drops sharply for: **Preference Inference** and **Preference Transfer**

	Multi-turn										**Single-turn**									
	Pref. Recall			**Pref. Inference**			**Pref. Transfer**			**Avg.**	**Pref. Recall**			**Pref. Inference**			**Pref. Transfer**			**Avg.**
LLM Backbone	P-EM	EA-F1	**OA-F1**	P-EM	EA-F1	**OA-F1**	P-EM	EA-F1	**OA-F1**	**OA-F1**	Prec.	Rec.	**F1**	Prec.	Rec.	**F1**	Prec.	Rec.	**F1**	**F1**
LLM																				
CodeGemma-7B	18.67	38.88	38.17	4.10	32.78	30.35	0.64	37.19	29.37	32.63	19.63	67.31	30.39	12.53	54.27	20.36	5.00	15.04	7.50	19.42
Gemma-3-12B	7.23	60.36	49.49	2.73	57.64	48.16	0.00	55.86	46.22	46.95	47.78	38.78	42.81	43.24	38.23	40.58	13.65	8.47	10.46	32.66
R1-Distill-Llama-7B	34.94	65.12	61.03	18.43	62.60	58.02	6.14	59.37	49.57	56.21	32.29	71.47	44.48	25.24	70.65	37.20	8.13	18.01	11.21	30.96
R1-Distill-Qwen-8B	13.55	33.49	31.58	7.17	27.88	25.50	0.64	25.87	20.12	25.73	21.12	56.51	30.75	13.33	44.37	20.51	3.10	8.26	4.51	18.59
GPT-4o-mini	32.23	58.21	53.54	18.43	62.46	57.34	4.87	61.98	48.94	53.27	50.09	76.18	60.44	42.39	78.84	55.13	16.10	27.12	20.21	45.26
GPT-5-mini	47.59	65.38	66.69	23.21	63.46	61.78	11.65	61.09	52.25	60.24	61.42	**88.64**	72.56	44.67	<u>81.57</u>	57.73	19.95	<u>36.02</u>	25.68	51.99
Gemini-3-Flash	<u>62.65</u>	72.73	<u>74.25</u>	28.67	69.66	<u>66.49</u>	14.62	69.68	56.54	<u>65.76</u>	63.27	<u>87.81</u>	73.55	44.32	81.23	57.35	22.11	33.69	26.70	52.53
GPT-5	51.20	62.33	64.77	<u>32.42</u>	65.34	64.01	**23.94**	64.27	55.47	61.42	59.39	86.70	70.50	43.22	76.11	55.13	19.25	31.36	23.85	49.83
Average	30.98	56.31	53.53	14.68	53.78	49.66	5.51	53.01	43.29		42.23	69.53	50.71	32.25	61.26	38.52	12.58	20.94	15.18	

Main Results

Baseline Observation : preference reasoning with full dialogue history

RAG & Mem0

- Retrieval-based approaches and summarization-centric memory modules fail to consistently improve performance on **Inference** and **Transfer** cases

→ *simple recall-based reasoning is feasible*

→ *but abstract or cross-domain preference reasoning remains challenging*

	Multi-turn										Single-turn									
	Pref. Recall			**Pref. Inference**			**Pref. Transfer**			**Avg.**	**Pref. Recall**			**Pref. Inference**			**Pref. Transfer**			**Avg.**
LLM Backbone	P-EM	EA-F1	**OA-F1**	P-EM	EA-F1	**OA-F1**	P-EM	EA-F1	**OA-F1**	**OA-F1**	Prec.	Rec.	**F1**	Prec.	Rec.	**F1**	Prec.	Rec.	**F1**	**F1**
LLM																				
Average	30.98	56.31	53.53	14.68	53.78	49.66	5.51	53.01	43.29		42.23	69.53	50.71	32.25	61.26	38.52	12.58	20.94	15.18	
RAG (Top-5)	23.19	65.01	56.06	23.55	62.42	56.62	15.47	64.52	54.13	55.60	45.07	59.56	51.31	42.98	70.99	53.54	17.60	26.69	21.21	42.02
Mem0	44.88	42.78	46.83	5.12	15.85	15.60	1.06	14.76	12.85	25.09	54.44	13.57	21.73	46.39	15.36	23.08	27.45	5.93	9.76	18.19

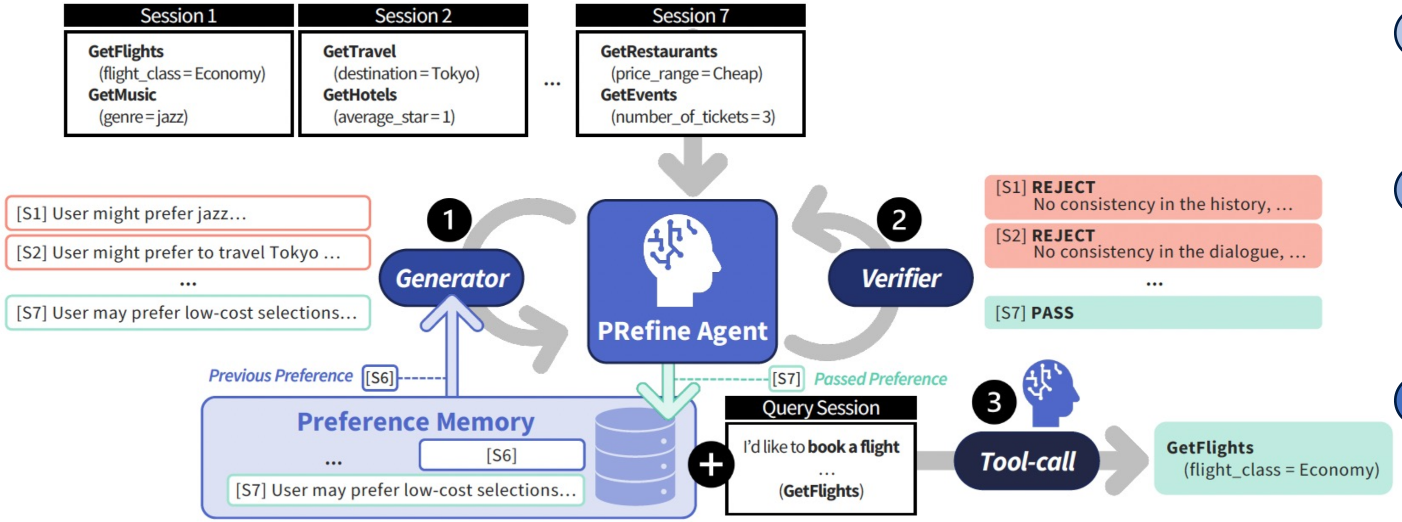
Preference reasoning cannot be solved by retrieval or summarization−based memory alone

Method Overview: PRefine

PRefine: a test-time preference reasoning framework

- *Assumption*: User **preferences** should be modeled as **latent hypotheses** that
  - emerge from **repeatedly observed choices** across interaction history,
  - and must be **continuously accumulated** and evaluated over time.

*Latent **Pre**ference **Refine**ment Memory* ; **PRefine Agent**



Generate
- propose candidate preference hypotheses from behavioral evidence

Verify
- evaluate hypotheses for consistency across sessions & domains
- reject over-specific or unsupported hypotheses

Refine
- retain only abstractions supported by repeated evidence
- update memory incrementally as new interactions arrive

PRefine treats preferences as evolving **hypotheses** rather than static facts.

Main Results

Effect of PRefine: consistent improvements in preference reasoning

PRefine (ours)

- **PRefine** consistently improves both preference identification and tool-calling quality
  - **Preference Recall** and Inference gains are more pronounced in smaller models
  - **Preference Transfer** gains are larger in stronger backbone models

→ *Improved preference understanding contributes to better tool-call execution.*

LLM Backbone	Multi-turn										Single-turn									
	Pref. Recall			Pref. Inference			Pref. Transfer			Avg.	Pref. Recall			Pref. Inference			Pref. Transfer			Avg.
	P-EM	EA-F1	**OA-F1**	P-EM	EA-F1	**OA-F1**	P-EM	EA-F1	**OA-F1**	**OA-F1**	Prec.	Rec.	**F1**	Prec.	Rec.	**F1**	Prec.	Rec.	**F1**	**F1**
PREFINE																				
CodeGemma-7B	59.64	69.50	70.51	16.38	65.86	61.00	1.61	67.20	53.97	61.83	35.40	81.22	49.31	30.51	70.65	40.80	7.41	18.43	10.57	33.56
Gemma-3-12B	20.48	**79.28**	69.27	5.67	**74.11**	63.24	0.21	75.38	63.54	65.35	**76.10**	63.66	69.30	52.10	57.54	54.28	12.67	6.36	8.45	44.01
R1-Distill-Llama-7B	42.05	62.35	61.63	22.12	62.07	58.22	4.83	52.22	42.95	54.27	44.72	71.30	54.95	28.82	60.68	39.08	9.26	13.77	11.07	35.03
R1-Distill-Qwen-8B	32.17	59.05	54.60	17.20	58.93	51.20	3.60	47.38	37.81	47.87	36.00	57.23	44.19	26.69	49.15	34.58	10.88	16.74	13.18	30.65
GPT-4o-mini	49.88	72.65	68.71	28.12	70.73	65.03	9.19	69.97	56.99	63.58	62.11	66.70	64.25	50.22	73.99	59.78	20.92	23.05	21.84	48.62
GPT-5-mini	51.45	68.03	68.08	**32.97**	67.71	65.16	21.02	67.23	58.47	63.90	73.23	83.43	**77.90**	**53.18**	76.72	62.79	29.59	30.00	29.62	56.77
Gemini-3-Flash	**64.88**	72.76	**74.75**	29.76	69.98	**67.17**	18.81	**70.55**	**59.62**	**67.18**	71.45	85.37	77.75	51.10	**82.05**	**62.95**	**30.92**	**39.87**	**34.81**	**58.50**
Avg. Gain (%p)	15.37	18.77	18.02	7.41	19.00	16.65	2.96	17.20	14.90		20.20	6.38	16.17	14.40	6.06	13.40	6.72	1.38	4.78	
Average	45.79	69.09	66.79	22.75	67.06	61.57	8.47	64.28	53.34		57.00	72.70	62.52	41.80	67.25	50.61	17.38	21.17	18.50	

Case Study

Effect of PRefine: verification rejects specificity and rewards consistency

Verification acts as a gatekeeper that **filters out** brittle hypotheses
and **retains** only abstractions supported by consistent behavioral evidence.

Session	API Calls	Action	Description
S1	GetMovies(average_rating=6);	Draft	Moderately rated movies inferred as a preference.
		Verify	[REJECT] Over-specific and unsupported abstraction.
		Refine	Generalized to accessible movie content.
		Verify	[REJECT] Insufficient evidence for future decisions.
		Refine	Reduced to minimal interest in movies.
		Verify	[PASS] Abstract and observation-supported.
S2	GetWeather(city=San Francisco);	Draft	Movie-centered preference maintained from prior session.
		Verify	[REJECT] Failed to account for weather-domain interaction.
		Refine	Prioritize movies while allowing other domains.
		Verify	[PASS] Cross-domain flexibility ensured.
S3	GetRentalCars(car_type = Standard); GetRestaurants(price_range = cheap);	Draft	Economical and simple options selected across domains.
		Verify	[PASS] Consistent cross-domain behavioral signal.
S4	GetHotels(average_star = 1);	Draft	Budget-friendly and simple interaction preference formulated.
		Verify	[PASS] Stable and memory-worthy preference.
PREFINE Memory		Budget-conscious and straightforward interaction style.	
[Inference Example] Query: “I’d like to book a flight.” → Inference: GetFlights(flight_class = Economy)			

Contribution

1 Preference Reasoning Benchmark for Tool-Calling

- Introduce the **first tool-calling benchmark** explicitly designed for preference reasoning
- Formalize three reasoning types: **Preference Recall**, **Inference**, and **Transfer**

2 Preferences as Action-Level Constraints

- Re-define preferences as **constraints over API arguments**, rather than surface-level statements or summarized dialogue context
- Enable **action-level selection** guided by inferred user preferences

3 Verifier-Guided Preference Abstraction

- Propose a test-time memory framework (**PRefine**) that performs verifier-guided preference abstraction
- Treat preferences as **latent hypotheses** that are verified/refined/rejected over time

4 Analysis of Preference-Aware Tool Use

- Provide extensive analysis of how preference reasoning impacts downstream tool execution quality
- Show consistent improvements across multi-session and cross-domain settings

Thank You

Yejin Yoon

HYU NLP Lab.

Dept. of Computer Science
Hanyang University, South Korea

stillwithyou@hanyang.ac.kr