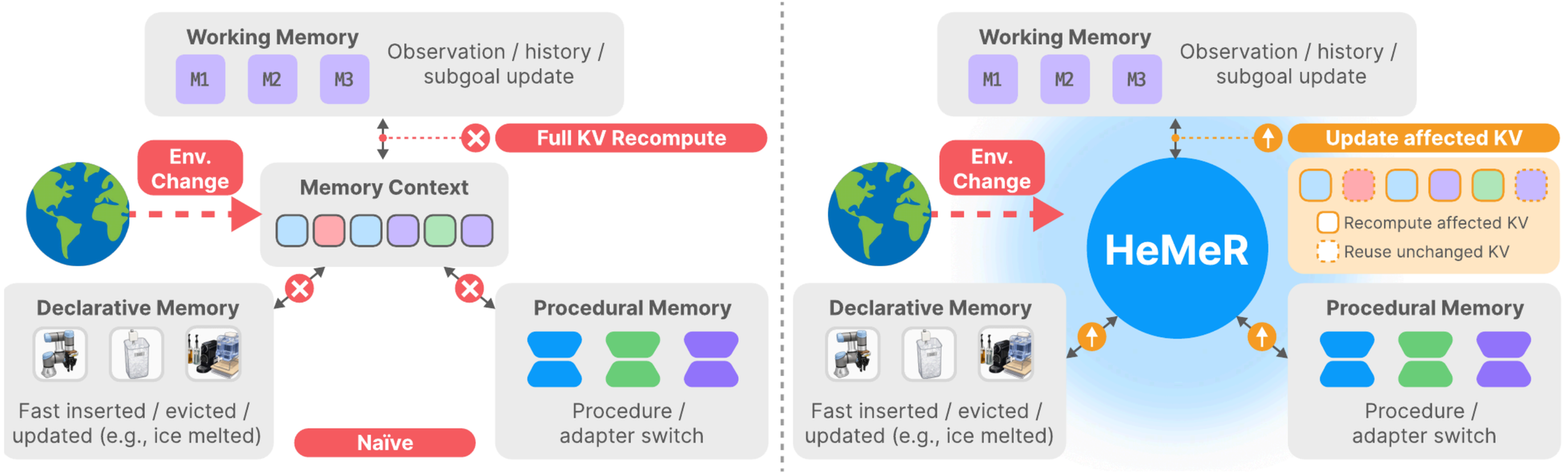
HeMeR

Heterogeneous Memory Reconciliation for Embodied Agents via Structured KV Reuse

Saehun Chun, Wonje Choi, Jinwoo Jang, Taeyoon Kwack, Honguk Woo

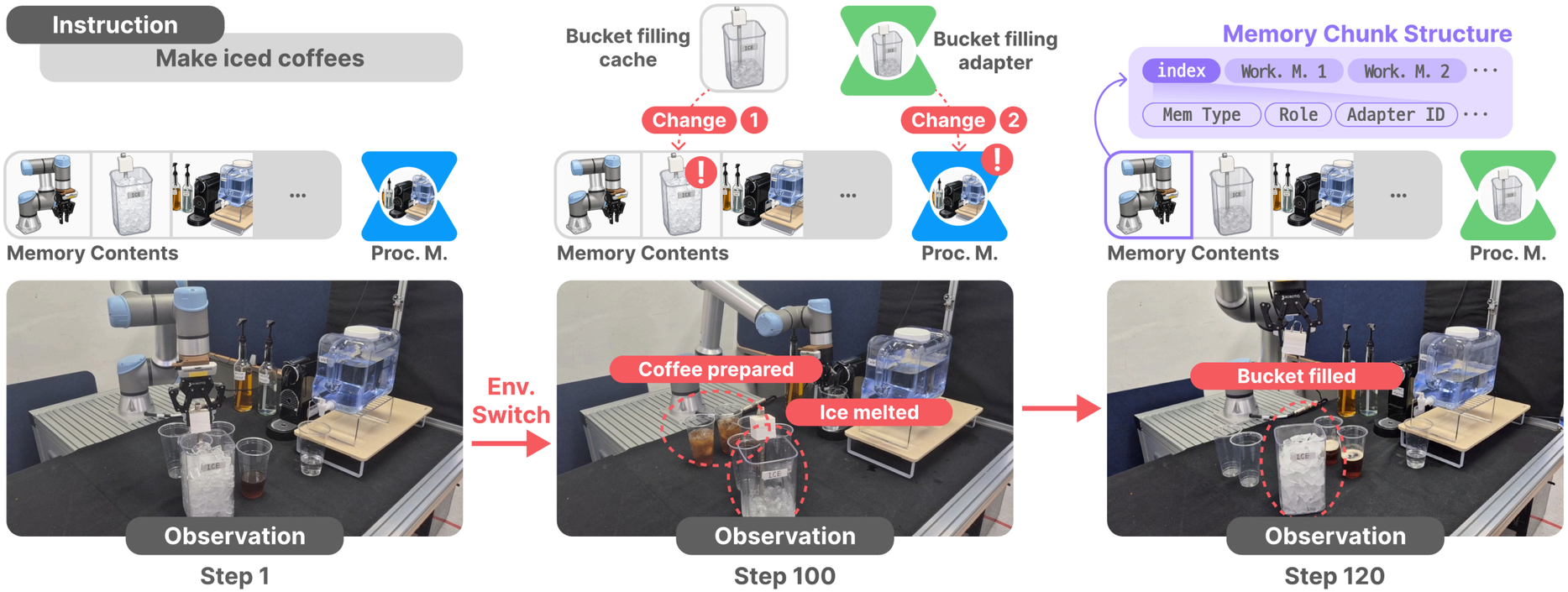
Sungkyunkwan University · NeurIPS 2026

Problem: Memory Changes Every Step



- Declarative + Procedural + Working → one **memory context** → KV cache
- A small change (e.g., ice melted) → naive serving **recomputes the full KV cache**

Idea: Memory as Indexed Chunks



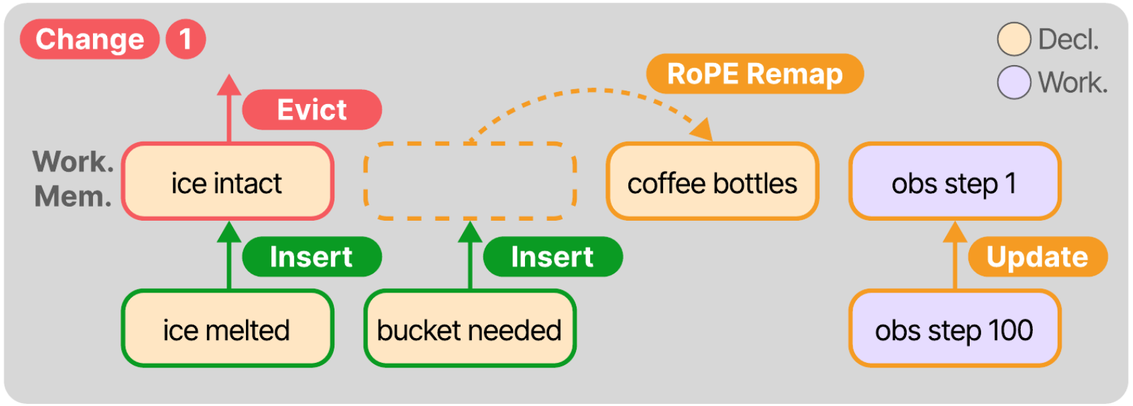
① Chunk-Level KV Reconciliation

Declarative & working memory updates

② Adapter-Stable KV Reconciliation

Procedural memory (LoRA adapter) switches

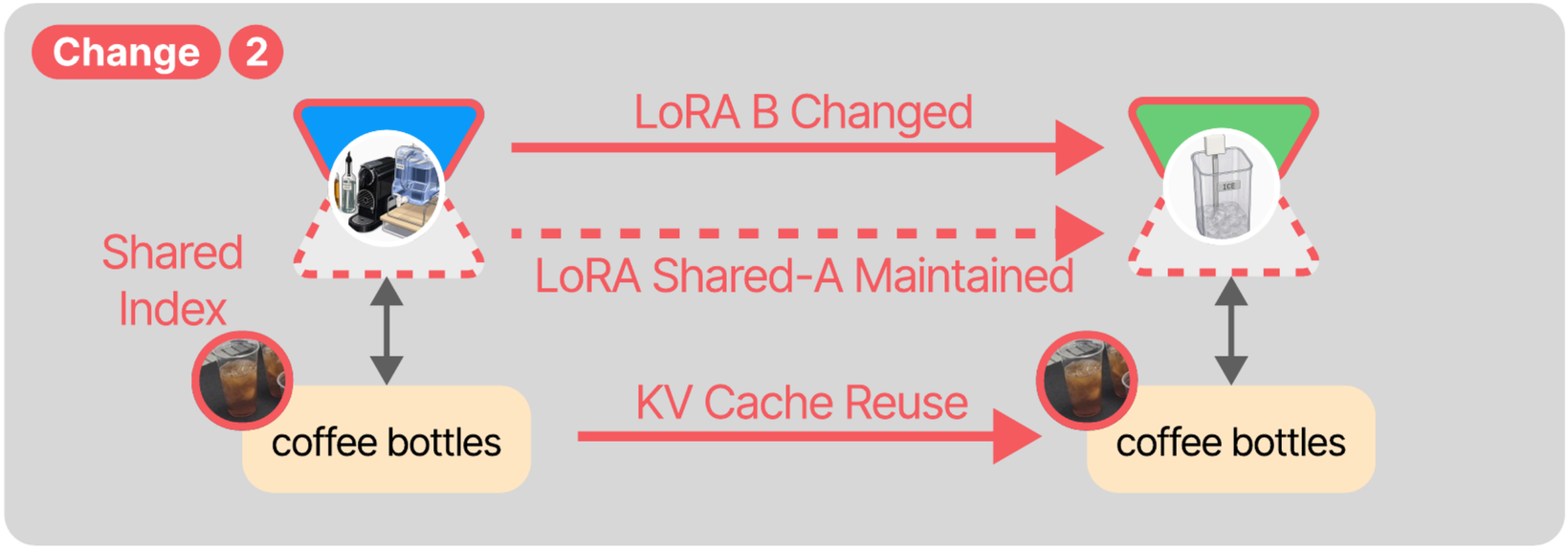
① Chunk-Level KV Reconciliation



Chunk change	KV action
Unchanged	**Reuse**
Moved only	**Reuse** + RoPE remap
New	Prefill (insert)
Removed	Evict
Changed	Recompute + deps

Match chunks by **ID** → check the **schema** → update only the **affected KV ranges**

② Adapter-Stable KV Reconciliation



Problem

A skill switch changes the active LoRA adapter. Old KV may not fit.

Check

The **adapter ID** is part of the KV reuse check.

Action

Recompute only the **skill chunk + dependents**. Shared-A LoRA keeps the rest reusable.

Results: Higher Success, Lower Latency

HeMeR vs. **MemRL**, the strongest of 6 baselines in success rate

ALFWorld

text · Llama-3.2-3B

SUCCESS RATE

71.4%

+18.6%p

LATENCY (S / STEP)

0.90 → **0.18**

5.0× faster

Habitat

vision · Qwen3-VL-4B

SUCCESS RATE

76.3%

+14.5%p

LATENCY (S / STEP)

2.68 → **1.82**

1.5× faster

Real Robot (UR7e)

vision · Qwen3-VL-4B

GOAL-COND. SUCCESS

83.2%

+18.9%p

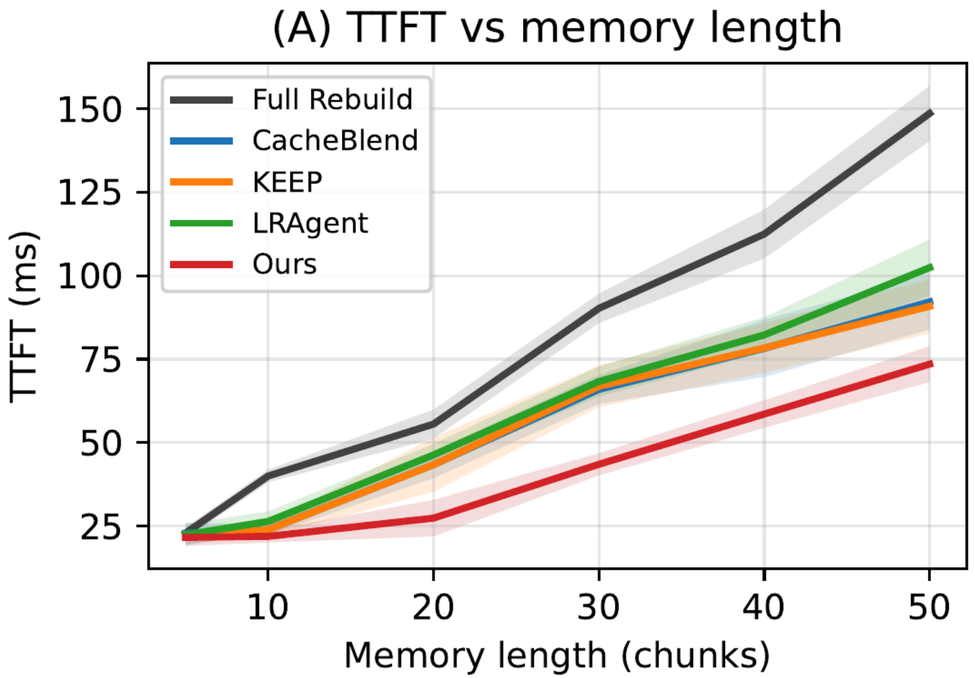
LATENCY (S / STEP)

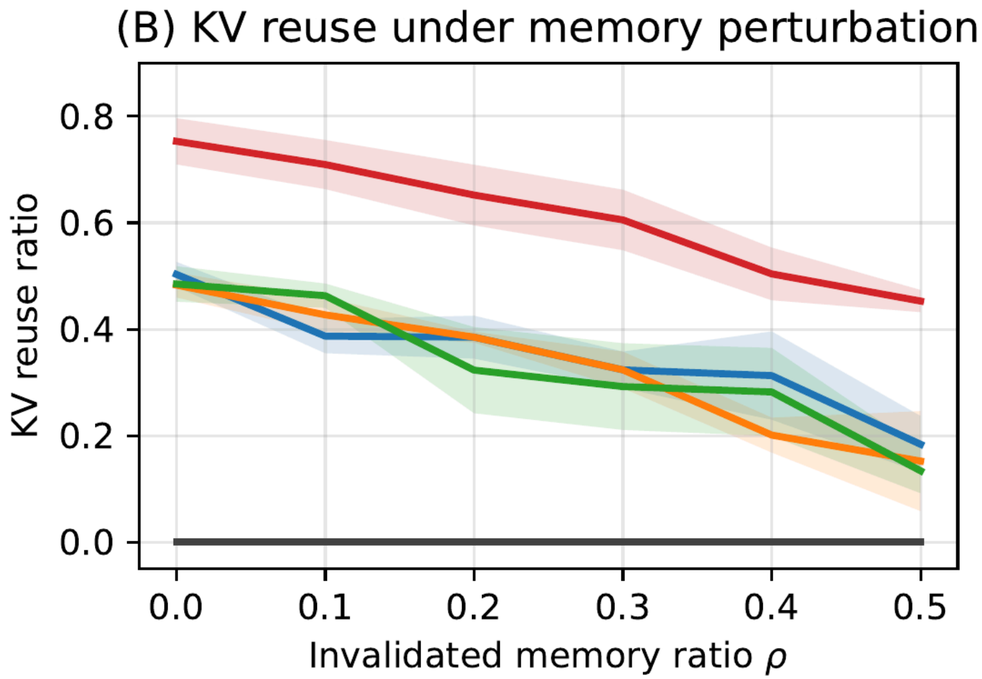
2.94 → **1.87**

1.6× faster

6 baselines: ReAct · Reflexion · MemRL · CacheBlend · KEEP · LRAgent — HeMeR is best in success, latency, and KV reuse in all three settings

Why Faster? Reuse What Is Still Valid





78.3%
of tokens reused (ALFWorld replay)

5.6%
of Full Rebuild's tokens recomputed

199.4 s → 25.8 s
total cache-processing time

Takeaways

- Memory updates in embodied agents = a **KV reconciliation** problem
- HeMeR → reuse valid KV, recompute only what changed
- Best success **and** latency on ALFWorld, Habitat, and a real robot

> *Next: multi-agent systems with shared memory (concurrent updates)*

Thank You

HeMeR · Questions?