CPGR-Mem: Cross-Platform Graph-based Routing Memory for Multimodal Intelligent Agents

1st Jiale Kong
School of Computer Science
Central China Normal University
Wuhan, China
jialekong2004@mails.ccnu.edu.cn

2nd Yuqi Zhao*
School of Computer Science
Central China Normal University
Wuhan, China
yuqizhao@ccnu.edu.cn

***Abstract*—Robust long-horizon execution in multimodal agents is severely hampered by inefficient cross-platform experience transfer. Existing memory mechanisms struggle to balance platform adaptability with the structured representation required for reliable reasoning. To address this, we introduce CPGR-Mem, a Cross-Platform Graph-based Routing Memory framework. CPGR-Mem organizes knowledge into a novel tri-graph architecture representing high-level strategies, platform-aware routing, and fine-grained execution traces. By employing a learnable routing mechanism and bidirectional retrieval, it dynamically constructs transferable task paths across heterogeneous environments. Evaluated on the KGCE benchmark across three leading multimodal backbones, CPGR-Mem significantly outperforms state-of-the-art baselines. Most notably, it achieves a 91.3% task completion ratio while reducing the backtracking rate by over 40%, offering a highly scalable, model-agnostic solution for cross-platform automation.**

***Index Terms*—Intelligent agents, cross-platform memory, graph-based routing, multimodal large language models, task execution**

I. INTRODUCTION

Multimodal Large Language Models (MLLMs) [1]–[3] have significantly advanced intelligent agents by unifying perception and reasoning across heterogeneous environments (e.g., desktop, web, mobile) [4]. However, executing robust long-horizon tasks across these platforms remains a formidable challenge [5]. While agents can perceive cross-platform contexts, they struggle to retain, structure, and transfer task knowledge effectively, leading to inefficient adaptation and execution failures in unseen environments [6].

Existing memory-augmented agents face a fundamental trade-off [7]. Platform-specific architectures (e.g., Voyager [8]) achieve high performance through tailored memory loops but overfit to specific APIs, rendering them brittle in heterogeneous settings [9]. Conversely, generalized semantic memories (e.g., Generative Agents [10]) offer flexibility via textual reuse but rely on shallow, unstructured representations. They fail to capture explicit structural relationships and semantic alignments across platforms, resulting in unstable long-horizon planning and poor adaptation [11]. Consequently, a critical gap persists: current agents lack a structural mechanism to systematically route and align transferable knowledge across heterogeneous boundaries.

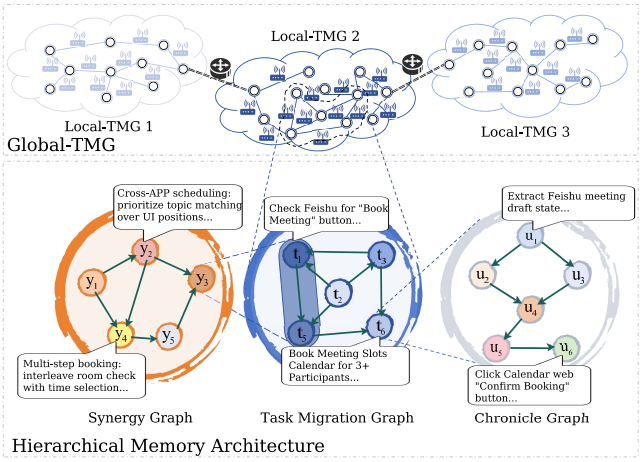

Fig. 1: Overview of CPGR-Mem's hierarchical memory architecture. The framework consists of three interconnected components: Synergy Graph for cross-platform strategies, Task Migration Graph for task-level knowledge routing, and Chronicle Graph for fine-grained execution traces.

To bridge this gap, we propose **CPGR-Mem** (Cross-Platform Graph-based Routing Memory). Inspired by network routing [12], CPGR-Mem re-frames cross-platform knowledge transfer as a structured graph routing problem. Designed as a plug-and-play module, it decouples memory management from the agent's core reasoning pipeline. As illustrated in Fig. 1, CPGR-Mem empowers agents to explicitly model task knowledge, dynamically navigate past experiences, and seamlessly adapt to novel interfaces, thereby ensuring coherent long-horizon execution.

In summary, our key contributions are threefold:

- **A Structured Cross-Platform Memory Architecture:** We propose CPGR-Mem, pioneering a hierarchical tri-graph system (Synergy Graph, Task Migration Graph, and Chronicle Graph) that elegantly formalizes the separa-

*Corresponding author.

This work is supported by the Postdoctoral Fellowship Program of China Postdoctoral Science Foundation (No. GZC20240571 and 2024M751052) and the National Natural Science Foundation of China (Nos. 62032016 and 61972292).

tion and integration of strategic, tactical, and operational knowledge across platforms.

- **Dynamic Routing and Bidirectional Retrieval:** We introduce a learnable routing mechanism over the Task Migration Graph to construct transferable experience paths. Coupled with a bidirectional retrieval strategy, it systematically bridges high-level strategic planning with actionable low-level execution steps.
- **State-of-the-Art Empirical Performance:** Extensive evaluations on the KGCE benchmark demonstrate that CPGR-Mem achieves model-agnostic superiority. Across three diverse MLLM backbones, it delivers up to a 91.3% task completion rate and reduces backtracking by over 40%, establishing a highly scalable solution for cross-platform UI automation.

II. RELATED WORK

A. Memory Mechanisms in Autonomous Agents

Existing memory-augmented agents generally fall into two extremes. **Platform-specific memory** closely couples action policies with environment-native representations (e.g., Voyager [8] in Minecraft, DOM-cues in WebArena [9]). While yielding high in-domain efficiency [13], these methods overfit to specific APIs and layouts, rendering them brittle and non-portable in unseen platforms [14]. Conversely, **unstructured semantic memory** (e.g., Generative Agents [10], MemoryBank [15], Mem0 [16]) stores experiences in a unified textual or embedding space to prioritize flexibility. However, these flat representations lack explicit structural guidance. Without hierarchical organization, agents struggle to decompose high-level cross-platform goals into actionable sequences, leading to hallucinated plans and inefficient retrieval [17]. CPGR-Mem directly bridges this divide by injecting explicit structural routing into generalized semantic spaces.

B. Structured Reasoning and Cross-Platform Transfer

To overcome the limitations of flat semantic memory, recent works introduce **structured representations** (e.g., hierarchical graphs) for intra-task optimization (H-Mem [18], Mem1 [19]) or multi-agent collaboration (MetaGPT [20], G-Memory [21]). Yet, these architectures remain inwardly focused on single-platform execution, lacking native mechanisms for inter-platform knowledge routing. Concurrently, research on **cross-platform transfer** (e.g., AgentGym [22]) highlights the necessity of multi-environment generalization. However, most efforts focus on evaluation benchmarks, system-level hardware optimization [23], [24], or simple few-shot adaptation algorithms [11], [25], rather than overhauling the agent's core memory architecture.

Unlike prior arts that treat structured memory and cross-domain transfer as isolated problems, CPGR-Mem intrinsically couples them. By formalizing a *vertically integrated, routable graph architecture*, we make cross-platform knowledge navigation a first-class citizen, filling the void left by current isolated and unstructured approaches.

III. PRELIMINARIES AND PROBLEM FORMULATION

Let $\mathcal{P} = \{p_1, p_2, \ldots, p_N\}$ denote a set of N heterogeneous platforms (e.g., Windows, Web, Android). During cross-platform execution, the agent accumulates operational traces into a unified memory. CPGR-Mem formalizes this cross-platform experience reuse as a routing problem over a hierarchical tri-graph architecture, defined as a tuple $\mathcal{G} = (\mathcal{G}_{\text{syn}}, \mathcal{G}_{\text{tmg}}, \mathcal{G}_{\text{chr}})$, which encapsulates strategic, tactical, and operational knowledge, respectively.

a) Synergy Graph.: $\mathcal{G}_{\text{syn}} = (\mathcal{V}_{\text{syn}}, \mathcal{E}_{\text{syn}})$ encodes high-level, transferable heuristics. Each node $v_k^{\text{syn}} = (r_k, \omega_k, \mathcal{P}_k) \in \mathcal{V}_{\text{syn}}$ represents a strategic rule r_k, an empirical confidence weight $\omega_k \in \mathbb{R}^+$, and its applicable platform scope $\mathcal{P}_k \subseteq \mathcal{P}$. Edges in $\mathcal{E}_{\text{syn}}$ capture semantic associations between strategies across different task contexts, providing abstract guidance prior to execution.

b) Task Migration Graph.: As the core routing layer, the Task Migration Graph (TMG) models task-step dependencies and cross-platform navigability:

$$\mathcal{G}_{\text{tmg}} = \left(\mathcal{V}_{\text{tmg}},\ \mathcal{E}_{\text{intra}} \cup \mathcal{E}_{\text{inter}}\right). \tag{1}$$

The vertex set $\mathcal{V}_{\text{tmg}} = \bigcup_{p \in \mathcal{P}} \mathcal{V}_{\text{tmg}}^{(p)}$ aggregates platform-specific task states, where each $v_i^{\text{tmg}} \in \mathcal{V}_{\text{tmg}}^{(p)}$ encapsulates a step description and its state snapshot. The edges are strictly partitioned: (1) **Intra-platform edges** $\mathcal{E}_{\text{intra}}$ denote local causal and temporal dependencies. (2) **Inter-platform edges** $\mathcal{E}_{\text{inter}}$ connect semantically aligned steps across different platforms, with each inter-edge $e_{ij} \in \mathcal{E}_{\text{inter}}$ assigned a learnable weight $\lambda_{ij} \in [0, 1]$ to quantify migration feasibility. Based on this edge partition, we decouple the TMG into two functional routing planes: the **Local-TMG** (governed by $\mathcal{E}_{\text{intra}}$) for guiding precise intra-platform execution, and the **Global-TMG** (governed by $\mathcal{E}_{\text{inter}}$) for facilitating dynamic cross-platform pathfinding.

c) Chronicle Graph.: $\mathcal{G}_{\text{chr}} = (\mathcal{V}_{\text{chr}}, \mathcal{E}_{\text{chr}})$ serves as an episodic memory recording fine-grained execution traces. A node $v_q^{\text{chr}} = (a_q, o_q, \tau_q, p_q) \in \mathcal{V}_{\text{chr}}$ logs an atomic action a_q, observation o_q, timestamp τ_q, and platform p_q. Edges $\mathcal{E}_{\text{chr}}$ represent strict temporal sequences. Crucially, we maintain a surjective mapping function $f: \mathcal{V}_{\text{chr}} \to \mathcal{V}_{\text{tmg}}$, anchoring low-level operation traces to high-level task nodes in the TMG. This enables bidirectional retrieval between planning and execution.

IV. METHODOLOGY

The CPGR-Mem framework formalizes cross-platform memory as a hierarchical routing and retrieval problem, as detailed in Fig. 2. Instead of treating past experiences as a flat pool of unstructured text, we hypothesize that an agent's "wisdom" should be stratified: high-level strategies guide the direction, tactical maps navigate the platforms, and operational traces execute the steps. The framework operates through a synergistic pipeline of path routing, bidirectional infusion, and adaptive evolution.

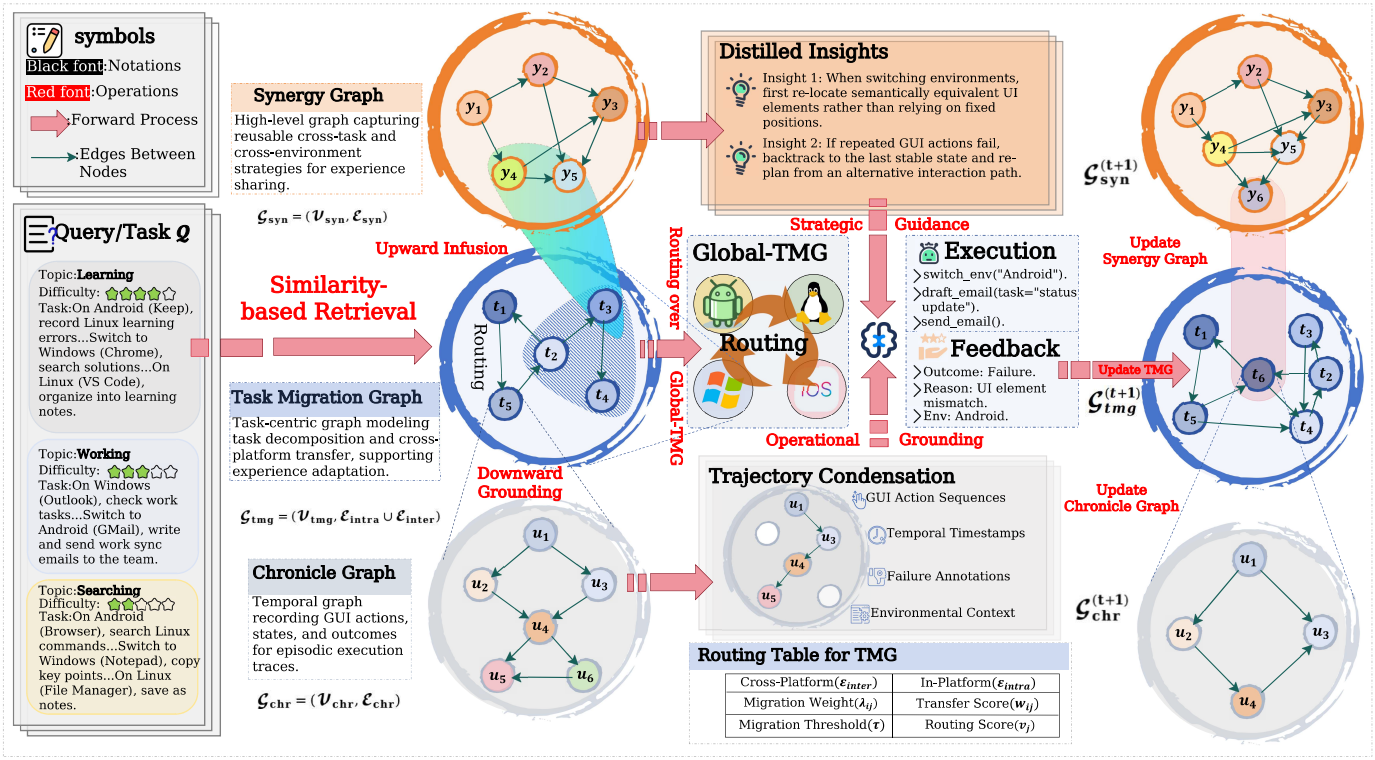

Fig. 2: The CPGR-Mem framework overview. Given a cross-platform task query, the system performs similarity-based retrieval, routes through the Task Migration Graph with upward/downward traversal, executes actions, and updates memory graphs based on feedback. For visual clarity, nodes in the Synergy Graph, Task Migration Graph, and Chronicle Graph are denoted as y_i, t_i, and u_i, respectively, corresponding to the formal definitions of v_i^{syn}, v_i^{tmg}, and v_i^{chr}.

A. Cross-Platform Routing via Task Migration Graph

The Global-TMG acts as the "routing plane" for the agent. When faced with a cross-platform task T, the agent navigates the graph to identify the most reliable experience path.

a) Entry Anchor Identification.: The process begins at the source platform p_s. The agent projects the current task T into a shared embedding space $\mathbf{z}$. We identify a set of entry anchors $\mathcal{V}_{\text{cand}} \subseteq \mathcal{V}_{\text{tmg}}^{(p_s)}$ by calculating the cosine similarity between the task embedding $\mathbf{z}_T$ and vertex embeddings $\mathbf{z}_{v_i}$:

$$\mathcal{V}_{\text{cand}} = \arg\text{top-}K_{v_i \in \mathcal{V}_{\text{tmg}}^{(p_s)}} \frac{\mathbf{z}_T \cdot \mathbf{z}_{v_i}}{\|\mathbf{z}_T\| \|\mathbf{z}_{v_i}\|}, \tag{2}$$

where we set $K = 5$ to ensure a diverse set of initial task hypotheses.

b) Dynamic Pathfinding and Migration.: From an anchor v_i, the agent expands a routing path $\mathcal{S}$ by evaluating the "cost" and "reward" of potential next hops. The transition to v_j is determined by a routing score that balances task relevance with historical reliability:

$$Score(v_j) = \alpha \cdot \text{sim}(\mathbf{z}_T, \mathbf{z}_{v_j}) + (1 - \alpha) \cdot w_{ij}, \tag{3}$$

where $\alpha = 0.6$ weights the current intent against prior experience. Crucially, when an inter-platform edge $e_{ij} \in \mathcal{E}_{\text{inter}}$ is encountered, it triggers a **Migration Protocol**. The agent evaluates the learnable weight λ_{ij}, which bridges the "semantic chasm" between platforms (e.g., from Windows CLI to Android UI). A migration is executed only if $Score(v_j) > \tau$ (with threshold $\tau = 0.75$), forming a robust cross-platform backbone.

B. Bidirectional Hierarchical Memory Retrieval

Once a path $\mathcal{S}$ is routed, CPGR-Mem "fleshes out" each tactical node with multi-level context through a bidirectional infusion mechanism.

a) Strategic Upward Infusion.: For each node $v_i^{\text{tmg}} \in \mathcal{S}$, the agent "looks up" to the Synergy Graph $\mathcal{G}_{\text{syn}}$ to retrieve strategic heuristics v_k^{syn} applicable to the platform $p(v_i)$. These nodes provide platform-invariant "wisdom" (e.g., "Always verify the UI state before scrolling"), ensuring the agent's behavior remains consistent with high-level objectives.

b) Operational Downward Grounding.: Simultaneously, the agent "looks down" to the Chronicle Graph $\mathcal{G}_{\text{chr}}$ via the surjective mapping $f^{-1}(v_i^{\text{tmg}})$. This retrieves the $M = 10$ most relevant episodic traces v_q^{chr}, providing "muscle memory" of specific action sequences and their historical outcomes. This bidirectional flow fuses abstract strategy with concrete execution, effectively mitigating the "hallucination" of unexecutable plans.

C. Execution-Driven Adaptive Update

CPGR-Mem evolves through a feedback loop, ensuring the memory substrate remains dynamic and performant.

TABLE I. Performance Comparison of Different Memory Mechanisms Across Multimodal Backbone Models

Model	Method	CR	CPA	Precision	Recall	F1-Score	BR	MSH
GPT-4o	No-memory	64.07$_{\uparrow 0.00}$	8.51$_{\uparrow 0.00}$	58.70$_{\uparrow 0.00}$	67.67$_{\uparrow 0.00}$	49.81$_{\uparrow 0.00}$	49.51$_{\uparrow 0.00}$	43.27$_{\uparrow 0.00}$
	Voyager	79.00$_{\uparrow 14.93}$	16.00$_{\uparrow 7.49}$	53.20$_{\downarrow 5.50}$	77.00$_{\uparrow 9.33}$	62.10$_{\uparrow 12.29}$	28.90$_{\downarrow 20.61}$	39.10$_{\downarrow 4.17}$
	MemoryBank	86.00$_{\uparrow 21.93}$	14.10$_{\uparrow 5.89}$	59.20$_{\uparrow 0.50}$	70.50$_{\uparrow 2.83}$	63.60$_{\downarrow 20.31}$	29.50$_{\downarrow 20.01}$	36.50$_{\downarrow 6.77}$
	Generative	75.30$_{\uparrow 11.23}$	12.20$_{\uparrow 3.69}$	63.00$_{\uparrow 4.30}$	64.50$_{\downarrow 3.17}$	61.90$_{\uparrow 12.09}$	17.30$_{\downarrow 32.21}$	38.00$_{\downarrow 5.27}$
	MetaGPT	76.70$_{\uparrow 12.63}$	10.80$_{\uparrow 2.29}$	61.40$_{\uparrow 2.70}$	77.80$_{\uparrow 10.13}$	68.10$_{\downarrow 18.29}$	32.80$_{\downarrow 16.71}$	42.80$_{\downarrow 0.47}$
	ChatDev	76.30$_{\uparrow 12.23}$	12.30$_{\uparrow 3.79}$	55.00$_{\downarrow 3.70}$	72.30$_{\uparrow 4.63}$	62.10$_{\uparrow 12.29}$	19.80$_{\downarrow 29.71}$	47.60$_{\uparrow 4.33}$
	CPGR-Mem	91.30$_{\uparrow 27.23}$	22.40$_{\uparrow 13.89}$	74.00$_{\uparrow 15.30}$	88.00$_{\uparrow 20.33}$	79.60$_{\uparrow 29.79}$	7.20$_{\downarrow 42.31}$	16.70$_{\downarrow 26.57}$
Qwen-VL	No-memory	52.31$_{\uparrow 0.00}$	5.77$_{\uparrow 0.00}$	45.60$_{\uparrow 0.00}$	56.18$_{\uparrow 0.00}$	47.80$_{\uparrow 0.00}$	53.65$_{\uparrow 0.00}$	40.78$_{\uparrow 0.00}$
	Voyager	60.30$_{\uparrow 7.99}$	6.50$_{\uparrow 0.73}$	35.20$_{\downarrow 10.40}$	63.35$_{\uparrow 7.17}$	43.90$_{\downarrow 3.90}$	39.00$_{\downarrow 14.65}$	28.60$_{\downarrow 12.18}$
	MemoryBank	64.60$_{\uparrow 12.29}$	9.70$_{\uparrow 3.93}$	34.08$_{\downarrow 11.52}$	57.76$_{\uparrow 1.58}$	39.60$_{\downarrow 8.20}$	24.60$_{\downarrow 29.05}$	52.30$_{\downarrow 11.52}$
	Generative	64.30$_{\uparrow 11.99}$	9.60$_{\uparrow 3.83}$	39.00$_{\downarrow 6.60}$	68.00$_{\uparrow 11.82}$	48.00$_{\uparrow 0.20}$	43.00$_{\uparrow 10.65}$	46.00$_{\uparrow 5.22}$
	MetaGPT	61.50$_{\uparrow 9.19}$	9.60$_{\uparrow 3.83}$	32.00$_{\downarrow 13.60}$	59.00$_{\uparrow 2.82}$	38.50$_{\downarrow 9.30}$	45.30$_{\downarrow 8.35}$	70.70$_{\uparrow 29.92}$
	ChatDev	65.00$_{\uparrow 12.69}$	7.60$_{\uparrow 1.83}$	33.70$_{\downarrow 11.90}$	50.70$_{\downarrow 5.48}$	43.50$_{\downarrow 4.30}$	35.20$_{\downarrow 18.45}$	69.00$_{\uparrow 28.22}$
	CPGR-Mem	89.00$_{\uparrow 36.69}$	15.90$_{\uparrow 10.13}$	64.47$_{\uparrow 18.87}$	85.90$_{\uparrow 29.72}$	72.08$_{\uparrow 24.28}$	13.30$_{\downarrow 40.35}$	16.60$_{\downarrow 24.18}$
Gemini 2.0 Flash	No-memory	61.80$_{\uparrow 0.00}$	7.14$_{\uparrow 0.00}$	52.67$_{\uparrow 0.00}$	65.91$_{\uparrow 0.00}$	66.89$_{\uparrow 0.00}$	39.81$_{\uparrow 0.00}$	40.91$_{\uparrow 0.00}$
	Voyager	73.60$_{\uparrow 11.80}$	8.60$_{\uparrow 1.46}$	63.40$_{\uparrow 10.73}$	77.00$_{\uparrow 11.09}$	68.70$_{\uparrow 1.81}$	20.76$_{\downarrow 19.05}$	32.50$_{\downarrow 8.41}$
	MemoryBank	73.90$_{\uparrow 12.10}$	10.80$_{\uparrow 3.66}$	65.50$_{\uparrow 12.83}$	79.20$_{\uparrow 13.29}$	71.10$_{\uparrow 4.21}$	36.90$_{\downarrow 2.91}$	44.20$_{\uparrow 3.29}$
	Generative	66.13$_{\uparrow 4.33}$	10.50$_{\uparrow 3.36}$	53.10$_{\uparrow 0.43}$	76.50$_{\uparrow 10.59}$	61.40$_{\uparrow 5.49}$	30.60$_{\downarrow 9.01}$	34.80$_{\downarrow 6.11}$
	MetaGPT	68.80$_{\uparrow 7.00}$	9.60$_{\uparrow 2.26}$	52.30$_{\downarrow 0.37}$	70.90$_{\uparrow 4.99}$	59.37$_{\downarrow 7.52}$	23.70$_{\downarrow 16.11}$	44.00$_{\uparrow 3.09}$
	ChatDev	71.80$_{\uparrow 10.00}$	9.30$_{\uparrow 2.16}$	65.30$_{\uparrow 12.63}$	84.18$_{\uparrow 18.27}$	72.60$_{\uparrow 5.71}$	24.30$_{\downarrow 15.51}$	37.20$_{\downarrow 3.71}$
	CPGR-Mem	93.80$_{\uparrow 32.00}$	22.30$_{\uparrow 15.16}$	71.90$_{\uparrow 19.23}$	90.40$_{\uparrow 24.49}$	79.10$_{\uparrow 12.21}$	3.50$_{\downarrow 36.31}$	14.30$_{\downarrow 26.61}$

Note: We highlight the best and second best results. All metrics are reported in percentages (%). ↑/↓ indicate change relative to the *No-memory* baseline. **Abbreviations:** CR: Completion Ratio; CPA: Completeness per Action; BR: Backtracking Rate; MSH: Max Steps Hit.

a) Weight Evolution.: Post-task execution, the success indicator $S \in \{0, 1\}$ is back-propagated to refine the TMG. We update the migration weights λ_{ij} using an exponential moving average (EMA) to ensure stability:

$$\lambda_{ij} \leftarrow (1 - \eta)\lambda_{ij} + \eta S, \tag{4}$$

where $\eta = 0.1$ is the learning rate. This reinforcement mechanism ensures that successful cross-platform "shortcuts" are prioritized in future routing.

b) Knowledge Consolidation.: Finally, new task steps are instantiated as $v_{\text{new}}^{\text{tmg}}$, and recurrent operational patterns are abstracted into new strategic nodes $v_{\text{new}}^{\text{syn}}$ in $\mathcal{G}_{\text{syn}}$. This completes the cycle from **Execution** to **Feedback**, and ultimately to **Wisdom Consolidation**, allowing the agent to grow more proficient with every cross-platform journey.

V. EXPERIMENTS

This section systematically evaluates the effectiveness and generality of the CPGR-Mem mechanism, focusing on the following research questions:

- **RQ1 (Effectiveness):** Does CPGR-Mem consistently outperform state-of-the-art memory agents in cross-platform task completion and execution efficiency?
- **RQ2 (Generalization):** Can CPGR-Mem maintain robust performance gains across diverse MLLM backbones (GPT-4o, Qwen-VL, Gemini 2.0)?
- **RQ3 (Ablation):** What are the individual contributions of the hierarchical graph routing and bidirectional retrieval mechanisms to the overall cross-platform stability?

A. Experimental Setup

a) Datasets and Benchmarks.: We evaluate CPGR-Mem on the **KGCE benchmark** [26], which features 120+ long-horizon tasks requiring seamless knowledge transfer between Android and Windows. To provide a holistic evaluation, we adopt seven standard metrics: (1) **Success Metrics:** Completion Ratio (CR) measures overall task fulfillment; (2) **Trajectory Quality:** Precision, Recall, and F1-Score assess the alignment of action sequences with gold-standard paths, while Completeness per Action (CPA) evaluates step-level granularity; (3) **Efficiency and Stability:** Backtracking Rate (BR) quantifies redundant exploration, and Maximum Steps Hit (MSH) indicates the frequency of task stagnation or timeout. This comprehensive suite ensures a rigorous assessment of both macro-success and micro-execution robustness.

b) Baselines and Backbones.: We compare CPGR-Mem against five representative agent paradigms: (1) **Voyager** [8]

TABLE II. Ablation Analysis of CPGR-Mem: Impact of Routing Strategy and Bidirectional Retrieval

Method	CR	CPA	Precision	Recall	F1-Score	BR	MSH
Routing Mechanism Ablation							
w/o Global-TMG Routing	$76.32_{\downarrow 12.68}$	$11.32_{\downarrow 4.58}$	$51.20_{\downarrow 13.27}$	$76.87_{\downarrow 9.03}$	$64.10_{\downarrow 7.98}$	$28.90_{\uparrow 12.43}$	$30.86_{\uparrow 14.26}$
w/o Local-TMG Routing	$70.75_{\downarrow 18.25}$	$10.43_{\downarrow 5.47}$	$50.30_{\downarrow 14.17}$	$74.54_{\downarrow 11.36}$	$62.17_{\downarrow 9.91}$	$31.47_{\uparrow 18.17}$	$32.85_{\uparrow 16.25}$
w/o TMG-based Routing	$67.32_{\downarrow 21.68}$	$9.35_{\downarrow 6.55}$	$47.48_{\downarrow 16.99}$	$69.53_{\downarrow 16.37}$	$59.31_{\downarrow 12.77}$	$32.87_{\uparrow 19.57}$	$33.89_{\uparrow 17.29}$
Bidirectional Retrieval Ablation							
w/o Upward Retrieval	$86.73_{\downarrow 2.27}$	$13.84_{\downarrow 2.06}$	$62.38_{\downarrow 2.09}$	$83.46_{\downarrow 2.44}$	$69.54_{\downarrow 2.54}$	$19.87_{\uparrow 3.27}$	$21.43_{\uparrow 4.83}$
w/o Downward Retrieval	$81.43_{\downarrow 7.57}$	$12.89_{\downarrow 3.01}$	$61.38_{\downarrow 3.09}$	$81.65_{\downarrow 4.25}$	$68.74_{\downarrow 3.34}$	$20.97_{\uparrow 7.67}$	$22.08_{\uparrow 5.48}$
w/o Bidirectional Retrieval	$78.62_{\downarrow 10.38}$	$11.39_{\downarrow 4.51}$	$59.79_{\downarrow 4.68}$	$80.31_{\downarrow 20.33}$	$67.56_{\downarrow 4.52}$	$21.60_{\uparrow 8.30}$	$23.76_{\uparrow 7.16}$
CPGR-Mem (Full Model)	$89.00_{\downarrow 0.00}$	$15.90_{\downarrow 0.00}$	$64.47_{\downarrow 0.00}$	$85.90_{\downarrow 0.00}$	$72.08_{\downarrow 0.00}$	$13.30_{\uparrow 0.00}$	$16.60_{\uparrow 0.00}$

Note: Blue cells denote the full CPGR-MEM model. ↑ and ↓ indicate the percentage increase and decrease relative to the full model, respectively. All metrics are in %. Ablation groups are partitioned by gray headers.
Abbreviations: CR: Completion Ratio; CPA: Completeness per Action; BR: Backtracking Rate; MSH: Max Steps Hit.

(skill-based iterative learning), (2) **MemoryBank** [15] and (3) **Generative Agents** [10] (unstructured semantic memory), and (4) **MetaGPT** [20] with (5) **ChatDev** [27] (multi-agent collaborative frameworks). To demonstrate model-agnostic robustness, we employ **GPT-4o** [1], **Qwen-VL** [2], and **Gemini 2.0 Flash** [3] as reasoning backbones.

c) Implementation Details.: For memory graph construction, we utilize the `all-MiniLM-L6-v2` encoder ($d = 384$). In the routing mechanism, we empirically set the balance factor $\alpha = 0.6$ to weigh current intent against historical reliability, and the migration threshold $\tau = 0.75$ to safely gate cross-platform transitions based on preliminary validation. To ensure statistical significance, each task is executed across 5 independent trials, with the MLLM temperature fixed at 0.2.

B. Overall Performance Comparison (RQ1)

Key Finding: CPGR-Mem establishes a new state-of-the-art across all metrics, demonstrating superior execution accuracy and navigation efficiency in cross-platform scenarios.

As shown in Table I, CPGR-Mem consistently outperforms all baselines. Under GPT-4o, it achieves the highest Completion Ratio of 91.30%, surpassing MemoryBank by 5.30%. This advantage is even more pronounced with Qwen-VL, yielding a 36.69% absolute gain over the no-memory baseline and significantly outperforming Voyager. At the step level, CPGR-Mem records the highest Completeness per Action and F1-Score, indicating highly precise action alignment with optimal trajectories.

Crucially, CPGR-Mem resolves the reasoning loops common in existing agents. While baselines frequently exceed a 17% Backtracking Rate, our framework slashes this to 3.50% under Gemini 2.0. Combined with the lowest Maximum Steps Hit, these results confirm that the global routing mechanism effectively prevents redundant exploration and ensures stable cross-platform navigation.

C. Robustness Across Multimodal Backbones (RQ2)

Key Finding: CPGR-Mem delivers consistent and substantial performance gains across diverse multimodal backbones, demonstrating high model-agnostic robustness.

As summarized in Table I, the framework achieves the highest overall scores whether paired with GPT-4o, Gemini 2.0 Flash, or Qwen-VL. The advantage is particularly evident with Qwen-VL, where CPGR-Mem elevates the Completion Ratio to 89.00%, outperforming the strongest baseline by an absolute margin of 24 percentage points. Concurrently, it yields significantly higher Completeness per Action and F1-Scores while suppressing the Backtracking Rate to 13.30%. This indicates that our memory architecture effectively stabilizes long-horizon execution even when relying on varying levels of inherent model reasoning.

Such uniform improvements across heterogeneous architectures confirm that the performance leaps are strictly driven by the structured cross-platform routing and bidirectional memory retrieval mechanisms, rather than any model-specific heuristics.

D. Ablation Study (RQ3)

Key Finding: Both the hierarchical routing mechanism and bidirectional memory retrieval are indispensable. Ablating any single component severely degrades either cross-platform transition success or in-platform execution stability.

As detailed in Table II, we systematically dismantle CPGR-Mem to isolate the contribution of each module.

Impact of Graph Routing: Disabling the Global-TMG causes the Completion Ratio to plummet from 89.00% to 76.32%, confirming its vital role as the bridge for inter-platform knowledge transfer. Conversely, removing the Local-TMG inflicts a catastrophic rise in the Backtracking Rate, surging from 13.30% to 31.47%. This stark contrast proves a clear division of labor: global routing dictates platform transitions, while local routing prevents erratic exploration within a specific environment. When all TMG routing is removed, the framework suffers maximum degradation across all efficiency metrics.

Impact of Bidirectional Retrieval: The hierarchy of memory retrieval demonstrates asymmetric but complementary

importance. Disabling downward retrieval yields a severe 7.57% drop in task completion, underscoring that fine-grained operational traces are strictly necessary to ground high-level intents into executable actions. Ablating upward retrieval results in a smaller yet significant decline, indicating that strategic heuristics prevent the agent from losing long-term goal alignment. When both retrieval directions are severed, the agent rapidly succumbs to reasoning loops and maximum step limits.

VI. LIMITATIONS AND FUTURE WORK

While CPGR-Mem demonstrates significant improvements in cross-platform workflows, it presents several avenues for future refinement. First, the hierarchical tri-graph naturally expands as the agent accumulates execution traces. Over extended lifecycles, this monotonic growth may introduce retrieval latency and storage overhead. Future work will explore dynamic memory pruning and knowledge distillation to maintain long-term scalability. Second, the current inter-platform routing relies on empirically set semantic thresholds (e.g., $\tau = 0.75$). Adopting reinforcement learning to dynamically adjust these parameters based on real-time task complexity could further enhance adaptability. Third, our framework is inherently bottlenecked by the visual grounding capabilities of the backbone MLLMs; severe UI hallucinations by the base model can occasionally mislead the routing process. Finally, extending our empirical validation beyond Windows and Android to closed-source ecosystems (e.g., iOS) remains a compelling direction for comprehensive agent generalization.

VII. CONCLUSION

This paper introduces CPGR-Mem, a novel Cross-Platform Graph-based Routing Memory framework designed to overcome the knowledge transfer barriers in multimodal intelligent agents. By unifying a hierarchical tri-graph architecture with a learnable routing mechanism and bidirectional retrieval, CPGR-Mem seamlessly bridges high-level strategic planning with platform-specific execution. Extensive evaluations on the KGCE benchmark demonstrate that our framework establishes a new state-of-the-art across diverse MLLM backbones. It not only achieves significant breakthroughs in task completion but also drastically eliminates redundant exploration and reasoning loops. Ultimately, CPGR-Mem provides a robust, model-agnostic foundation for the development of scalable and generalist cross-platform autonomous agents.

REFERENCES

[1] A. Hurst *et al.*, "GPT-4o System Card," *arXiv preprint arXiv: 2410.21276*, 2024.

[2] J. Bai *et al.*, "Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond," *arXiv preprint arXiv:2308.12966*, 2023.

[3] G. Comanici *et al.*, "Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities," *arXiv preprint arXiv:2507.06261*, 2025.

[4] L. Wang *et al.*, "A Survey on Large Language Model based Autonomous Agents," *Front. Comput. Sci.*, vol. 18, no. 6, p. 186345, 2024.

[5] N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language Agents with Verbal Reinforcement Learning," in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 8634–8652.

[6] H. Shi *et al.*, "Continual Learning of Large Language Models: A Comprehensive Survey," *ACM Comput. Surv.*, vol. 58, no. 5, pp. 1–42, 2025.

[7] Z. Zhang *et al.*, "A Survey on the Memory Mechanism of Large Language Model-based Agents," *ACM Trans. Inf. Syst.*, vol. 43, no. 6, pp. 1–47, 2025.

[8] G. Wang *et al.*, "Voyager: An Open-Ended Embodied Agent with Large Language Models," *Trans. Mach. Learn. Res.*, 2023.

[9] S. Zhou *et al.*, "WebArena: A Realistic Web Environment for Building Autonomous Agents," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

[10] J. S. Park *et al.*, "Generative Agents: Interactive Simulacra of Human Behavior," in *Proc. ACM Symp. User Interface Softw. Technol. (UIST)*, 2023, pp. 1–22.

[11] X. Liu *et al.*, "AgentBench: Evaluating LLMs as Agents," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

[12] W. Kwon *et al.*, "Efficient Memory Management for Large Language Model Serving with PagedAttention," in *Proc. 29th ACM Symp. Oper. Syst. Princ. (SOSP)*, 2023, pp. 611–626.

[13] K.-L. Du, R. Zhang, B. Jiang, J. Zeng, and J. Lu, "Understanding Machine Learning Principles: Learning, Inference, Generalization, and Computational Learning Theory," *Mathematics*, vol. 13, no. 3, p. 451, 2025.

[14] A. Patel and V. Kant, "Enhanced Cross-Domain Recommendation System Using Collaborative Filtering and Transfer Learning Methods," *SN Comput. Sci.*, vol. 6, no. 6, p. 581, 2025.

[15] W. Zhong, L. Guo, Q. Gao, H. Ye, and Y. Wang, "MemoryBank: Enhancing Large Language Models with Long-Term Memory," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 19 724–19 731.

[16] P. Chhikara, D. Khant, S. Aryan, T. Singh, and D. Yadav, "Mem0: Building Production-ready AI Agents with Scalable Long-term Memory," *arXiv preprint arXiv:2504.19413*, 2025.

[17] M. Hu, T. Chen, Q. Chen, Y. Mu, W. Shao, and P. Luo, "HiAgent: Hierarchical Working Memory Management for Solving Long-horizon Agent Tasks with Large Language Model," in *Proc. 63rd Annu. Meet. Assoc. Comput. Linguist. (ACL)*, 2025, pp. 32 779–32 798.

[18] H. Sun and S. Zeng, "Hierarchical Memory for High-efficiency Long-term Reasoning in LLM Agents," *arXiv preprint arXiv:2507.22925*, 2025.

[19] Z. Zhou *et al.*, "Mem1: Learning to Synergize Memory and Reasoning for Efficient Long-horizon Agents," *arXiv preprint arXiv:2506.15841*, 2025.

[20] S. Hong *et al.*, "MetaGPT: Meta Programming for a Multi-agent Collaborative Framework," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

[21] G. Zhang, M. Fu, K. Wang, G. Wan, M. Yu, and S. Yan, "G-Memory: Tracing Hierarchical Memory for Multi-Agent Systems," in *Adv. Neural Inf. Process. Syst.*, vol. 38, 2025.

[22] Z. Xi *et al.*, "AgentGym: Evaluating and Training Large Language Model-based Agents Across Diverse Environments," in *Proc. 63rd Annu. Meet. Assoc. Comput. Linguist. (ACL)*, 2025, pp. 27 914–27 961.

[23] C. Zhang *et al.*, "JENGA: Effective Memory Management for Serving LLM with Heterogeneity," in *Proc. 31st ACM Symp. Oper. Syst. Princ. (SOSP)*, 2025, pp. 446–461.

[24] Y. Zhao *et al.*, "Cross-domain Recommendation via Adaptive Bi-directional Transfer Graph Neural Networks," *Knowl. Inf. Syst.*, vol. 67, no. 1, pp. 579–602, 2025.

[25] X. Jing and K. Qian, "Reducing Cross-sensor Domain Gaps in Tactile Sensing via Few-sample-driven Style-to-content Unsupervised Domain Adaptation," *Sensors*, vol. 25, no. 1, p. 256, 2025.

[26] Z. Liu, S. Liu, and Y. Zhao, "Kgce: Knowledge-Augmented Dual-Graph Evaluator for Cross-Platform Educational Agent Benchmarking with Multimodal Language Models," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*. IEEE, 2025, pp. 7586–7591.

[27] C. Qian *et al.*, "ChatDev: Communicative Agents for Software Development," in *Proc. 62nd Annu. Meet. Assoc. Comput. Linguist. (ACL)*, 2024, pp. 15 174–15 186.