HabitatAgent: An End-to-End Multi-Agent System for Housing Consultation

Hongyang Yang[1⋆], Yanxin Zhang[1], Yang She[3], Yue Xiao[2], Hao Wu[1], Yiyang Zhang[1], Jiapeng Hou[1], and Rongshan Zhang[1]

[1] Fangdongdong,
[2] Tsinghua University,
[3] Columbia University

Abstract. Housing selection is a high-stakes and largely irreversible decision problem. We study *housing consultation* as a decision-support interface for housing selection. Existing housing platforms and many LLM-based assistants often reduce this process to ranking or recommendation, resulting in opaque reasoning, brittle multi-constraint handling, and limited guarantees on factuality.

We present **HabitatAgent**, the first LLM-powered multi-agent architecture for end-to-end housing consultation. HabitatAgent comprises four specialized **agent roles**: **Memory**, **Retrieval**, **Generation**, and **Validation**. The **Memory Agent** maintains multi-layer user memory through internal stages for constraint extraction, memory fusion, and verification-gated updates; the **Retrieval Agent** performs *hybrid vector–graph retrieval* (GraphRAG); the **Generation Agent** produces evidence-referenced recommendations and explanations; and the **Validation Agent** applies multi-tier verification and targeted remediation. Together, these agents provide an auditable and reliable workflow for end-to-end housing consultation.

We evaluate HabitatAgent on **100 real user consultation scenarios** (**300** multi-turn question–answer pairs) under an *end-to-end correctness* protocol. A strong single-stage baseline (Dense+Rerank) achieves **75%** accuracy, while HabitatAgent reaches **95%**.

Keywords: AI Agents · Housing Selection · Multi-Agent Systems · Decision Support Systems · GraphRAG · Trustworthy AI

1 Introduction

Housing selection is a consequential and largely irreversible decision with high switching costs and strong path dependence. Unlike many reversible consumer choices, it involves substantial transaction frictions, limited liquidity, and pervasive information asymmetry. Homebuyers therefore must reason over heterogeneous and sometimes conflicting factors—including budget, location, accessibility, building quality, community services, and policy constraints—often under incomplete or noisy evidence.

⋆ Corresponding author: hy2500@columbia.edu. Hongyang Yang serves as the CTO of Fangdongdong.

We study *housing consultation* as a decision-support interface for housing selection, whereas most existing platforms still reduce it to *search and ranking* [4,8,12,2]. Conversational recommendation, LLM-based assistants, and multi-agent systems have improved multi-turn interaction, preference modeling, and task decomposition [9,19,10,14,16]. However, reliable housing consultation remains challenging in practice because many systems still rely on monolithic prompting, weak evidence grounding, and limited safeguards against factual and entity errors.

When framed as an end-to-end housing consultation problem, these limitations manifest as four concrete challenges:

1. **Evolving and under-specified user preferences.** Buyers often begin with vague goals and progressively refine constraints over multi-turn dialogue; the system must elicit latent needs, disambiguate intent, and maintain a consistent preference state across turns [9,19].
2. **Heterogeneous evidence and relational constraints.** Accurate consultation requires integrating heterogeneous sources (projects, transit, schools, policies, costs) and enforcing relational or hard constraints that cannot be satisfied by semantic similarity alone, motivating graph-aware retrieval [6,11].
3. **Opaque recommendations and ungrounded shortlist decisions.** Users need auditable comparisons and explicit rationales aligned with their priorities, rather than opaque top-k ranking outputs [4,8].
4. **High cost of factual and entity errors.** Misstated numbers, conflated entities, or hallucinated amenities can directly mislead irreversible decisions, requiring explicit verification and targeted correction [3].

To address these challenges, we propose **HabitatAgent**, a production-oriented multi-agent architecture for end-to-end housing consultation. HabitatAgent organizes four specialized roles—**Memory**, **Retrieval**, **Generation**, and **Validation**—into a closed-loop workflow for reliable decision support. The system is built on three key mechanisms: (i) *Verification-Gated Memory*, which prevents unverified information from contaminating long-term user state; (ii) *Adaptive Retrieval Routing*, which selectively invokes graph-constrained retrieval for relationally complex queries; and (iii) *Failure-Type-Aware Remediation*, which applies targeted recovery instead of naive regeneration when validation fails. On 100 real consultation scenarios (300 multi-turn Q&A pairs), HabitatAgent improves end-to-end accuracy from 75% (Dense+Rerank) to 95%.

Contributions. This work makes three contributions:

1. We formulate buyer-side housing consultation as an *end-to-end, high-stakes decision-support problem*, beyond ranking-only real-estate recommendation.
2. We present a closed-loop multi-agent architecture that couples *verification-gated memory*, *adaptive vector–graph retrieval routing*, and *failure-type-aware remediation* for reliable multi-turn, multi-constraint consultation.
3. We show on real consultation scenarios that this design substantially improves end-to-end correctness over strong dense-retrieval, graph-retrieval, and self-correction baselines.

2 Related Work

2.1 Conversational and Decision-Support Systems for Housing

Housing consultation differs from standard real-estate recommendations because it involves multi-turn preference refinement, heterogeneous evidence, and high-stakes trade-offs. Prior work on real-estate recommender systems mainly studies ranking, filtering, and preference matching [8]. These systems are useful for candidate generation, but they usually treat the task as search or recommendation rather than end-to-end consultation, and therefore provide limited support for iterative clarification, explicit evidence grounding, and post-generation verification.

2.2 LLMs and Agents in Real-Estate Applications

Recent work has explored LLMs in real-estate settings, including listing-oriented generation, domain-adapted decision support, and agent-based assistants [15,18,7]. These studies show the promise of LLMs in the domain, but most focus on a single capability—such as generation, appraisal, or assistant interaction—rather than a closed-loop consultation workflow that jointly manages memory, retrieval, generation, and validation.

2.3 Multi-Agent LLM Systems and Graph-Grounded Retrieval

More broadly, LLM-based multi-agent systems have emerged as a general paradigm for decomposing complex tasks into specialized roles [10,13,17,5]. These directions are highly relevant to housing consultation, where users often impose relational constraints such as transit access, school districts, or location dependencies. In addition, recent housing-related benchmarks and datasets further highlight the importance of grounded evidence and reliable decision support in this domain [1,18].

2.4 Gap and Positioning of This Work

Our work differs from prior studies in two ways. First, we focus on buyer-side housing consultation as an end-to-end decision-support problem rather than a ranking-only or generation-only task. Second, the main contribution is not a single isolated module, but a closed-loop architecture that couples three mechanisms: verification-gated memory updates, adaptive vector–graph retrieval routing, and failure-type-aware remediation. This combination is designed specifically for high-stakes, multi-constraint consultation settings where correctness, traceability, and recovery matter as much as recommendation relevance.

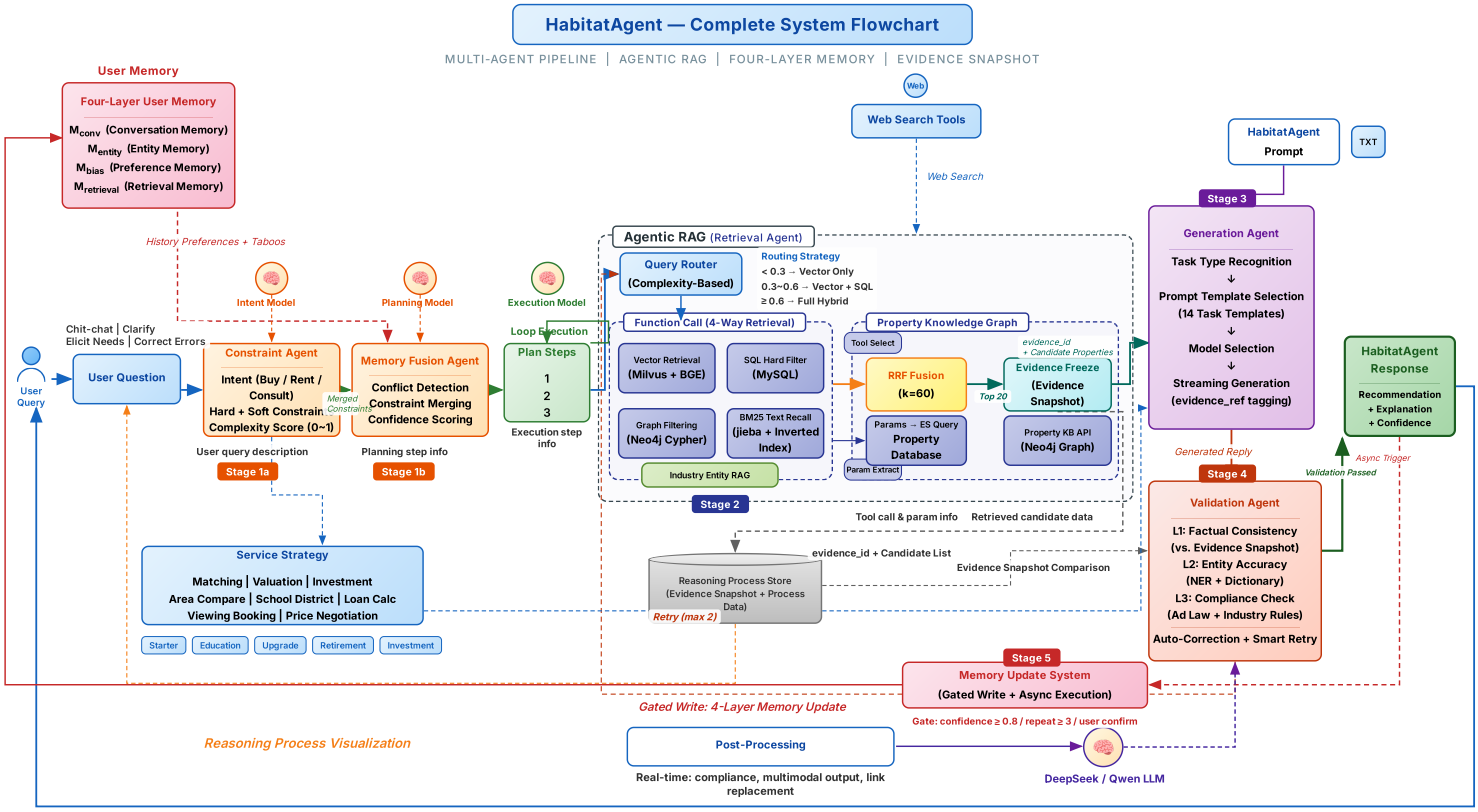

Fig. 1. HabitatAgent overview. Four specialized agents—**Memory**, **Retrieval**, **Generation**, and **Validation**—are executed as a **five-stage** workflow. In particular, Stage 1a (constraint extraction), Stage 1b (memory fusion), and Stage 5 (memory update) are internal substeps of the **Memory Agent**.

3 Methodology

HabitatAgent is a multi-agent architecture for end-to-end housing consultation. It organizes four specialized agents—**Memory**, **Retrieval**, **Generation**, and **Validation**—into a closed-loop workflow for reliable decision support. The design objective is not merely task decomposition, but robust decision support under evolving preferences, relational constraints, and high error cost.

Figure 1 summarizes the full workflow. At each turn, HabitatAgent first consolidates user state, then retrieves auditable evidence, generates a grounded response, and finally validates the output with targeted remediation when needed. The three methodological mechanisms studied in this paper—Verification-Gated Memory, Adaptive Retrieval Routing, and Failure-Type-Aware Remediation—are implemented across this workflow.

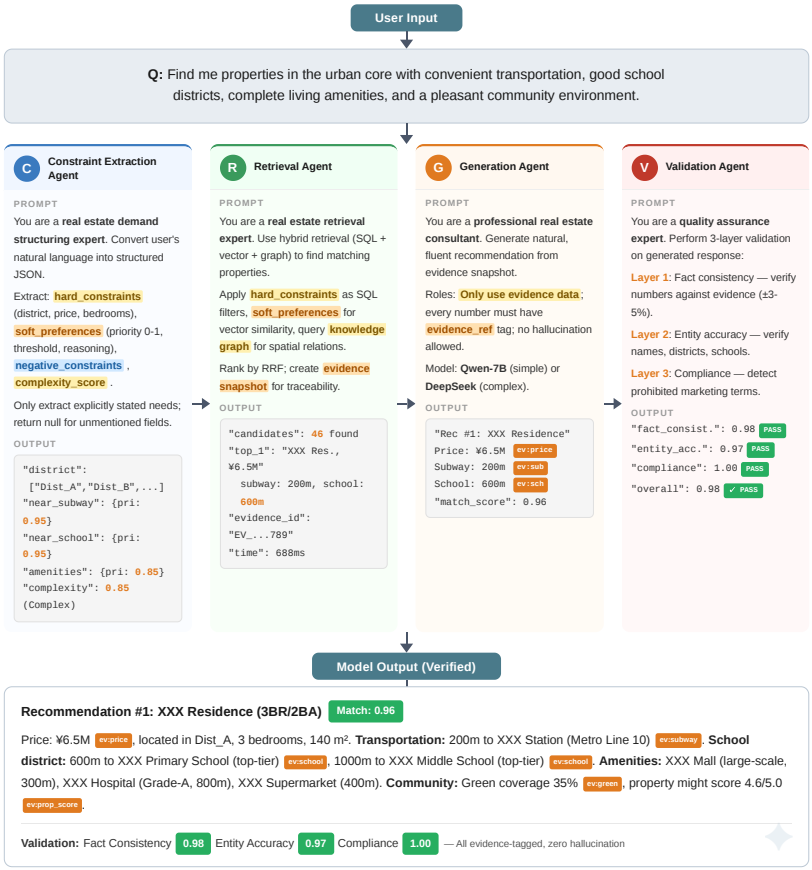

Fig. 2. Use case of HabitatAgent. A user query is processed through constraint extraction, hybrid retrieval (GraphRAG), evidence-grounded generation, and multi-tier validation to produce a verified recommendation.

Figure 2 further illustrates a concrete end-to-end example of how HabitatAgent processes a realistic housing query into a verified recommendation.

3.1 Memory Agent: Verification-Gated Multi-Layer Memory

Multi-turn housing consultation requires persistent preference tracking, yet naive memory accumulation can cause long-term drift due to polluted or unverified information. HabitatAgent addresses this tension via a **Verification-Gated Memory** mechanism: only information extracted from responses that pass multi-tier validation is allowed to update long-term user memory. This design prevents error propagation across turns and stabilizes personalization over time.

Four-layer memory structure The Memory Agent maintains a four-layer hierarchy:

- **Conversational memory ($M_{\mathbf{conv}}$)**: a short-term buffer storing the most recent 5 turns, with a TTL of 24 hours.
- **Entity memory ($M_{\mathbf{entity}}$)**: a mid-term store of referenced entities (e.g., properties, regions) represented as a weighted sorted list.
- **Bias memory ($M_{\mathbf{bias}}$)**: a long-term store of explicit preferences/aversions (e.g., "dislikes noisy areas"), each with a weight $w \in [-1, 1]$.
- **Retrieval memory ($M_{\mathbf{retrieval}}$)**: a 7-day rolling cache of previously recommended property IDs, used for deduplication and diversification.

Toward context-aware conversational memory. Our current conversational memory M_{conv} keeps the most recent 5 turns to control context length and latency. However, real housing consultation can involve long-horizon reference and backtracking (e.g., comparing a property mentioned much earlier). To mitigate context fragmentation, we optionally maintain a lightweight **context-aware state** that summarizes salient constraints and referenced entities, and updates this state at each turn (e.g., ADD/UPDATE/RELAX). This allows the Memory Agent to recover important earlier context without retaining the full raw dialogue.

Gated update policy Let $\text{extract}(Q_t, A_t)$ denote preference tuples extracted from the query–answer pair at turn t. The Memory Agent updates state only when the response A_t is validated:

$$M_{t+1} = \begin{cases} M_t \cup \{\text{extract}(Q_t, A_t)\}, & \text{if } V(A_t, R_t) = \text{pass} \\ M_t, & \text{otherwise} \end{cases} \tag{1}$$

where R_t denotes retrieved evidence used at turn t (cf. §3.2). The primary validation function focuses on factual and entity correctness:

$$V(A, R) = \begin{cases} \text{pass}, & \text{if } V_{\text{fact}}(A, R) \geq 0.85 \wedge V_{\text{entity}}(A, R) \geq 0.90 \\ \text{fail}, & \text{otherwise} \end{cases} \tag{2}$$

where V_{fact} checks factual consistency (e.g., price, size) and V_{entity} checks entity correctness (e.g., property names). In addition, we apply an auxiliary compliance filter $V_{\text{comp}}(A)$ as a conservative safety safeguard, but it is not part of the primary evaluation metric in §4.

3.2 Retrieval Agent: Adaptive Hybrid Vector–Graph Retrieval (GraphRAG)

Housing consultation requires evidence that is both semantically relevant and structurally consistent with hard/relational constraints (e.g., "near Line 10" or "within 30 minutes to CBD"). Pure dense retrieval is fast but may violate relational constraints; pure graph filtering is precise but expensive. HabitatAgent therefore employs a **hybrid vector–graph retrieval** design, and introduces an **Adaptive Retrieval Router** to decide when to invoke graph-constrained retrieval.

Adaptive retrieval router The router $f_\theta : Q \to [0, 1]$ predicts whether a query requires graph retrieval. It consumes a feature vector $\phi(Q)$ including:

- number of extracted constraints N_c,
- count of relational keywords (e.g., "near", "commute to") N_r,
- confidence drop in dense retrieval (e.g., $\text{score}_1 - \text{score}_5$),
- a binary history flag indicating whether similar queries previously failed (fail_h).

We optimize f_θ with a cost-sensitive loss, penalizing false negatives ($C_{\text{FN}} = 5$) more than false positives ($C_{\text{FP}} = 1$), since failing to route complex queries to graph retrieval is more harmful than paying extra retrieval cost.

Graph-constrained retrieval For queries routed as complex, the Retrieval Agent performs two-step hybrid retrieval: (i) dense retrieval returns a broad candidate set (top 100), then (ii) constraints are translated into a Cypher query executed over a property knowledge graph to filter candidates. Our graph contains 6,016 nodes (properties, regions, transit, schools, districts) and 45,000 edges encoding relations such as `LOCATED_IN` and `NEAR_SUBWAY`.

3.3 Generation Agent: Task-Aware Prompted Response Generation

Given structured constraints from the Memory Agent and evidence R_t from the Retrieval Agent, the Generation Agent produces **context-aware** and **evidence-referenced** responses through a task-aware orchestration layer. Instead of relying on a single prompt, it first identifies the task type of the current query and then selects one of **14 task-specific prompt templates** covering common housing consultation scenarios, including recommendation, property query, comparison, facility query, value analysis, investment, school-district consultation, first-time buyer guidance, second-hand housing, decoration, out-of-town purchase, short-term rental, policy interpretation, and general fallback.

This design enables a unified generation interface across diverse consultation intents while preserving evidence grounding. The Generation Agent then selects the appropriate model and produces a response with explicit evidence references for key factual claims, improving transparency and supporting downstream validation (§3.4).

3.4 Validation Agent: Multi-Tier Verification and Failure-Type-Aware Remediation

Because factual and entity errors can directly mislead high-stakes decisions, HabitatAgent verifies each candidate response before presenting it to the user and before committing updates to long-term memory. The Validation Agent applies **multi-tier checks** and triggers **failure-type-aware remediation** to recover from common failure modes instead of terminating.

Failure classification When validation fails, the agent labels the failure as one of:

- **Entity missing**: the response mentions an entity not supported by retrieved evidence.
- **Constraint conflict**: retrieved results cannot satisfy constraints simultaneously.
- **Factual error**: numeric/categorical claims contradict evidence.

Remediation policy Let V_{fail} denote the failure type. The remediation policy maps failures to targeted actions:

$$\text{Remediate}(V_{\text{fail}}) = \begin{cases} \text{RetrieveByEntity}(e_{\text{miss}}), & \text{if entity\_missing} \\ \text{RelaxThreshold}(\tau \to 0.9\tau), & \text{if constraint\_conflict} \\ \text{LocalCorrect}(A, V_{\text{issues}}), & \text{if fact\_error} \end{cases} \tag{3}$$

Entity-missing triggers a retrieval centered on e_{miss}; constraint-conflict relaxes the least important retrieval threshold and retries; factual-error performs local correction guided by identified issues, avoiding full regeneration. This closed-loop design increases the rate of *validated* responses and enables **verification-gated** memory updates in §3.1.

4 Evaluation

We evaluate HabitatAgent as an *auditable, end-to-end* housing consultation system. The design of our experiments follows the system logic introduced in §3: (i) **Memory** supports evolving preferences via verification-gated updates, (ii) **Retrieval** handles heterogeneous and relational constraints via adaptive hybrid vector–graph retrieval, (iii) **Generation** produces evidence-referenced recommendations and explanations, and (iv) **Validation** enforces factual/entity correctness with failure-type-aware remediation.

Our evaluation answers two research questions:

- **RQ1 (Overall Effectiveness):** Does HabitatAgent improve *end-to-end correctness* and recommendation quality over strong single-stage and graph-based baselines?
- **RQ2 (Component Contribution):** How much do **Verification-Gated Memory**, **Adaptive Retrieval Routing**, and **Failure-Type-Aware Remediation** each contribute to the final performance?

4.1 Experimental Setup

Data We use a proprietary dataset derived from real, anonymized user interactions on a Beijing housing platform.

Table 1. Overall performance comparison on the full dataset (300 queries). Best results are in **bold**.

System	Accuracy	nDCG@5	Faithfulness	P95 Latency (ms)
B1: Monolithic RAG	0.72	0.76	0.78	450
B2: Dense+Rerank	0.75	0.80	0.82	380
B3: GraphRAG-Fixed	0.82	0.85	0.88	820
B4: LLM-Ranker	0.70	0.78	0.75	1200
B5: Self-RAG	0.78	0.82	0.85	680
B6: Rule-Verifier	0.80	0.83	0.86	520
Ours: HabitatAgent	**0.95**	**0.92**	**0.96**	**720**

- **Property corpus.** 5,000 property listings with structured attributes (e.g., price, layout, area, geo-coordinates) and linked amenities (e.g., subway, schools). This corpus is used to construct both the vector index and the property knowledge graph.
- **Consultation scenarios.** 100 real consultation scenarios, each with a 3-turn dialogue, resulting in 300 user queries for end-to-end evaluation. Queries are categorized as *Simple* (80%, 1–2 constraints) or *Complex* (20%, $\geq$3 constraints and/or relational requirements).
- **Human annotations.** Three trained annotators label each query-response pair for constraint satisfaction and factual/entity correctness. Inter-annotator agreement is substantial (Cohen's $\kappa = 0.82$).

Ethics. All user data is anonymized and used with explicit consent for research purposes.

Baselines. We compare HabitatAgent against six baselines representing common design choices: **B1: Monolithic RAG**, a single-prompt dense-retrieval system without memory or verification; **B2: Dense+Rerank**, a dense retriever (BGE) with a reranker, but without memory or verification; **B3: GraphRAG-Fixed**, graph-based retrieval for all queries without adaptive routing; **B4: LLM-Ranker**, which ranks a large candidate set without explicit verification; **B5: Self-RAG**, which performs error detection followed by full regeneration; and **B6: Rule-Verifier**, a rule-based verifier that refuses responses when errors are detected. For a fair comparison, all methods use the same underlying LLM and, where applicable, the same candidate pool.

Metrics We report metrics that reflect both recommendation utility and consultation reliability:

- **Recommendation quality:** nDCG@5.
- **Constraint satisfaction: CSR@5** (Constraint Satisfaction Rate), the fraction of recommended items in top-5 that satisfy all *hard* constraints.
- **Grounded generation quality:** RAGAS **Faithfulness**.

Table 2. Performance on the complex query subset (60 queries with ≥3 constraints).

System	Accuracy	CSR@5	P95 Latency (ms)
B2: Dense+Rerank	0.62	0.08	380
B3: GraphRAG-Fixed	0.85	0.88	820
Ours: HabitatAgent	**0.95**	**0.95**	**680**

- **End-to-end accuracy (primary):** the percentage of responses that (i) satisfy all constraints, (ii) are factually correct with respect to retrieved evidence, and (iii) contain correct entity references.
- **System latency:** P95 end-to-end latency (ms).

4.2 Main Results

Overall performance (RQ1). Table 1 reports results on all 300 queries. HabitatAgent achieves the best end-to-end accuracy (0.95) and the highest grounded generation faithfulness (0.96), outperforming all baselines. Compared with the strongest accuracy baseline, GraphRAG-Fixed (0.82), HabitatAgent improves end-to-end accuracy by 13 percentage points while reducing P95 latency by 100 ms. This result suggests that reliable housing consultation requires more than stronger retrieval alone; it benefits from coupling persistent memory, evidence-grounded generation, and post-generation validation within a single workflow.

Complex queries and relational constraints (RQ1). Table 2 reports results on the subset of 60 complex queries with at least three constraints. Dense+Rerank suffers a substantial drop in CSR@5 (0.08), indicating that semantic similarity alone is insufficient for hard relational constraints. HabitatAgent maintains 0.95 CSR@5 and 0.95 end-to-end accuracy, while remaining faster than always-on graph retrieval. This suggests that adaptive routing is effective in preserving relational correctness without incurring the full cost of graph-constrained retrieval on every query.

4.3 Ablation Study

Component efficacy (RQ2) We conduct ablations corresponding to the three methodological mechanisms described in §3: **Verification-Gated Memory**, **Adaptive Retrieval Routing**, and **Failure-Type-Aware Remediation**. Table 3 reports end-to-end accuracy after removing each component.

Removing **Adaptive Retrieval Routing** causes the largest degradation ($0.95 \rightarrow 0.75$), indicating that selective graph-aware retrieval is critical for complex relational queries. This result suggests that graph-constrained retrieval should be invoked when query complexity requires it, rather than uniformly applied to all cases.

Table 3. Ablation study results (end-to-end accuracy).

Configuration	Accuracy	Change
Full System (HabitatAgent)	0.95	–
w/o Verification Gate	0.88	−7.0pp
w/o Adaptive Routing	0.75	−20.0pp
w/o Failure Remediation	0.87	−8.0pp
w/o Multi-Tier Validation	0.85	−10.0pp

Removing the **Verification Gate** (0.95 → 0.88) and **Failure Remediation** (0.95 → 0.87) also leads to substantial drops, showing that reliability depends not only on retrieval quality but also on validated memory updates and recovery from validation failures. Finally, removing **Multi-Tier Validation** reduces accuracy to 0.85, highlighting the importance of explicit factual and entity checks before returning responses and updating memory.

5 Conclusion

We presented HabitatAgent, a multi-agent architecture for end-to-end housing consultation. Rather than treating housing consultation as ranking or generation alone, HabitatAgent organizes memory, retrieval, generation, and validation into a closed-loop workflow for reliable decision support. Across 100 real consultation scenarios (300 multi-turn Q&A pairs), the proposed design improves end-to-end accuracy from 75% to 95% over a strong Dense+Rerank baseline.

More broadly, our findings suggest that in high-stakes, multi-constraint decision-support tasks, correctness depends not only on retrieval quality or model capability in isolation, but also on how memory updates, evidence access, response generation, and validation are coordinated. Future work will extend the evaluation to more cities, larger datasets, and dynamically updated knowledge graphs.

References

1. Anusha Bagalkotkar, Aveek Karmakar, Gabriel Arnson, and Ondrej Linda. Fairhome: A fair housing and fair lending dataset. *arXiv preprint arXiv:2409.05990*, 2024.
2. Michael Ball and V. Srinivasan. Using the analytic hierarchy process in house selection. *Journal of Real Estate Finance and Economics*, 9(1):69–85, 1994.
3. Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.
4. Alireza Gharahighehi, Konstantinos Pliakos, and Celine Vens. Recommender systems in the real estate market—a survey. *Applied Sciences*, 11(16):7502, 2021.

5. Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)*, 2024.
6. Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A. Rossi, Subhabrata Mukherjee, Xianfeng Tang, Qi He, Zhigang Hua, Bo Long, Tong Zhao, Neil Shah, Amin Javari, Yinglong Xia, and Jiliang Tang. Retrieval-augmented generation with graphs (GraphRAG). *CoRR*, abs/2501.00309, 2025.
7. Joakim Bruslund Haurum, Davide Pauli, et al. Real estate with AI: An agent based on LangChain, GPT and RAG for professional real estate market value estimation. *Procedia Computer Science*, 232:1745–1754, 2024.
8. Carlos Henríquez-Miranda, Jesús Ríos-Pérez, and Germán Sanchez-Torres. Recommender systems in real estate: A systematic review. *Bulletin of Electrical Engineering and Informatics*, 14(3):2156–2170, 2025.
9. Dietmar Jannach and Li Chen. Conversational recommendation: A grand AI challenge. *AI Magazine*, 43(2):151–163, 2022.
10. Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. A survey on LLM-based multi-agent systems: Workflow, infrastructure, and challenges. *Vicinagearth*, 1:9, 2024.
11. Microsoft. GraphRAG. `https://github.com/microsoft/graphrag`. GitHub repository. Accessed: 2026-02-25.
12. Harshit Oberoi, Anil Goyal, Nikhil Sikka, et al. Re-recsys: An end-to-end system for recommending properties in real-estate domain. *arXiv preprint arXiv:2404.16553*, 2024.
13. Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D. Nguyen. Multi-agent collaboration mechanisms: A survey of LLMs. *CoRR*, abs/2501.06322, 2025.
14. Zhefan Wang, Yuanqing Yu, Wei Zheng, Weizhi Ma, and Min Zhang. MACRec: A multi-agent collaboration framework for recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*, 2024.
15. Jibang Wu, Chenghao Yang, Yi Wu, Simon Mahns, Chaoqi Wang, Hao Zhu, Fei Fang, and Haifeng Xu. Ai realtor: Towards grounded persuasive language generation for automated copywriting. *arXiv preprint arXiv:2502.16810*, 2025.
16. Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. Fingpt: Open-source financial large language models. *arXiv preprint arXiv:2306.06031*, 2023. First official FinGPT paper; FinLLM Workshop at IJCAI 2023.
17. Hongyang Yang, Boyu Zhang, Neng Wang, Cheng Guo, Xiaoli Zhang, Likun Lin, Junlin Wang, Tianyu Zhou, Mao Guan, Runjia Zhang, et al. Finrobot: An open-source ai agent platform for financial applications using large language models. *arXiv preprint arXiv:2405.14767*, 2024.
18. Kexin Zhu and Yang Han. Real: Benchmarking abilities of large language models for housing transactions and services. *arXiv preprint arXiv:2507.03477*, 2025.
19. Yaochen Zhu, Harald Steck, Dawen Liang, Yinhan He, Nathan Kallus, and Jundong Li. LLM-based conversational recommendation agents with collaborative verbalized experience. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 2207–2220, Suzhou, China, 2025. Association for Computational Linguistics.