Knowledge-Enhanced Cobot Reasoning Assistant for Robot Manipulation

Ruibo Tang[1], Clara-Larissa Lorenz[1], Jérôme Frisch[1] and Christoph van Treeck[1]
[1]Institute of Energy Efficiency and Sustainable Building (E3D), RWTH Aachen University, Aachen, Germany
E-mail(s): ruibo.tang@rwth-aachen.de, lorenz@e3d.rwth-aachen.de, frisch@e3d.rwth-aachen.de, treeck@e3d.rwth-aachen.de

Abstract: The construction industry is exploring collaborative robots (cobots) and sensing technologies to support construction automation. With the emergence of large language models (LLMs), their integration with cobots could introduce a new paradigm in construction, where language serves as a direct interface for guiding and coordinating robotic assistance. However, language instruction control of cobots in dynamic environments remains challenging: (a) LLMs may generate actions that are plausible but not feasible, (b) long-horizon tasks may require interactive human feedback during execution, and (c) developing high-quality domain-specific knowledge bases remains costly and time-consuming. In this work, we present *Knowledge-Enhanced Cobot Reasoning Assistant* (***KECRA***), an LLM and Visual Foundation Models (VFMs) based framework. ***KECRA*** enables intuitive dialogue-based interaction with human operators, reducing the need for expert knowledge in programming or task specification. VFMs act as the eyes of ***KECRA***, perceiving target objects and capturing real-time spatial context to inform planning and execution. Core reasoning uses a state-graph agent (*Instruction* → *Plan Sub-tasks* → *Confirm Sub-tasks* → *Generate Path* → *Confirm Path*) with rewindable human checkpoints at key nodes for plan correction. All interactions are automatically accumulated into a knowledge graph, enabling LLMs retrieval-augmented reasoning. We validate ***KECRA***'s practicality and effectiveness in tabletop manipulation experiments.

Keywords: Collaborative Robots, Large Language Model, Multi-Modal, AI Agent, Knowledge Graph

DOI: 10.18154/RWTH-CONV-254921. Published in the conference proceedings of the 36. Forum Bauinformatik 2025, Aachen, Germany,

1 Introduction

Robotic applications in construction aim to address labor shortages and improve productivity, with collaborative robots (cobots) enabling safe human–robot interaction [1], [2]. To expand their flexibility and usability, more recent studies have focused on using LLMs, which offer zero-shot learning, commonsense reasoning, and contextual understanding capabilities, for robot manipulation [3], [4], [5], [6]. However, language instruction control of cobots in unstructured environments such as construction sites remains challenging: (1) LLMs may generate action sequences that are commonsensical but physically infeasible in the current scene due to hallucination [7], [8]; (2) end-to-end LLM planning

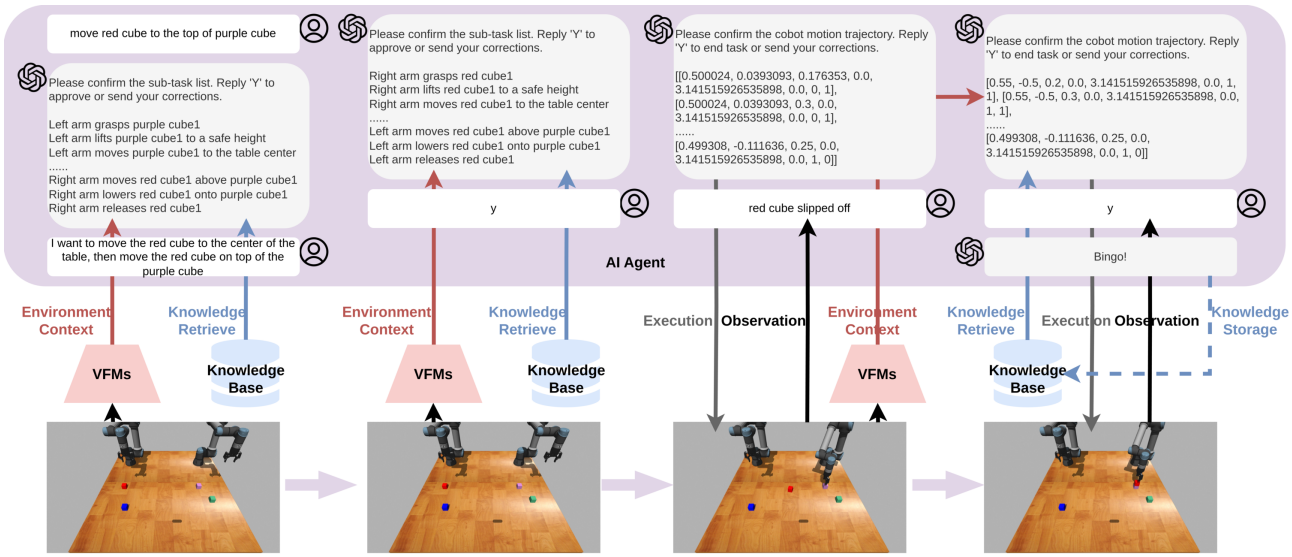

Figure 1: Overview of KECRA with example task "move red cube to the top of purple cube".

pipelines lack mid-execution feedback mechanisms, preventing timely error correction and introducing safety hazards [7], [9]; and (3) Retrieval-Augmented Generation (RAG) systems struggle to interpret implicit operational preferences in instructions [4]. A representative case is spatial command ambiguity, such as "place the object beside another", does "beside" mean to the left or to the right?

To address these challenges, we propose ***KECRA***, *Knowledge-Enhanced Cobot Reasoning Assistant*, which integrates a transparent loop state graph with human-in-the-loop (HIL) checkpoints to enable controllable and interpretable plan execution, and a structured knowledge graph that incrementally stores prior task interactions to support RAG. We show the code of ***KECRA*** at https://github.com/RWTH-E3D/KECRA. For example, when given the instruction “move the red cube to the top of the purple cube” (Figure 1), the high-level task planner integrates environmental context with prior task knowledge and decomposes the command into sub-tasks for operator approval. The operator revises the plan, prompting the agent to regenerate sub-tasks using the updated context and retrieved feedback examples. After validation, a low-level pose planner generates the trajectory and sends it to the robot arms for execution. When a slip occurs, the agent replans with the new context, completes the task, and logs the entire interaction to the knowledge base for future use.

The main contributions are: (1) a novel cobot assistant framework ***KECRA***, which integrates VFMs, a state-graph planner with human verification, and a self-enhancing knowledge graph to improve task success in dual-arm tabletop manipulation; (2) a pipeline that automatically converts human–AI interaction histories into structured knowledge graphs, enhancing planning efficiency while sustaining performance for complex instructions.

2 Related Work

LLM-powered robotic manipulation: LLMs with hundreds of billions of parameters, such as GPT-4o [10], have advanced language-conditioned robotic manipulation by enhancing in-context planning and reasoning capabilities [3]. Recent research on robotic manipulation leverages these foundation

models in three main directions: (1) *zero-shot action sequence generation*, where LLMs infer task steps directly from natural language [11]; (2) *HIL adaptation*, allowing real-time feedback to refine plans during execution [12], [13]; and (3) *direct code synthesis*, replacing predefined skill libraries with executable programs [4], [5]. In addition, a set of auxiliary modules are often integrated into LLM-based planners to ground task objectives and prevent hallucinations: (1) reward mechanisms to align generated plans with physical and safety constraints [7], [14]; (2) conformal prediction frameworks to resolve ambiguity through structured multiple-choice reasoning [13]; and (3) vision-language models (VLMs) to provide environmental grounding [5], [15]. However, most prior works implement either high-level task planning or low-level code generation in isolation [16], lacking ***KECRA***'s dual-phase integrated planning that ensures long-horizon tasks are executed in alignment with operator preferences. Moreover, corrective feedback is typically handled as ad hoc interventions [17], rather than through ***KECRA***'s rewindable state-graph workflow, which enables transparent backtracking to specific reasoning nodes when errors are detected. While VLMs and reward functions in previous works are often used as post hoc success detectors [18], ***KECRA*** leverages them as active causal context providers: real-time semantic perceptions (e.g., object dimensions, spatial relations) directly condition LLM reasoning, preventing physically infeasible plans and informing context-aware replanning.

Knowledge base for LLMs reasoning: LLMs with RAG rely on domain-relevant knowledge bases to retrieve indexed text corpora as contextual examples [19], typically using sentence embeddings for similarity matching [20]. In language-conditioned robotic manipulation, especially under multi-turn interaction scenarios, the design of the knowledge base fundamentally shapes how LLM outputs translate into robotic control [8], [21]. Most prior works ground LLM reasoning on one of the following knowledge sources: (1) static few-shot examples embedded in the prompt [5], [7], [21]; (2) rule summaries distilled from multi-turn feedback [6], [22]; (3) structured symbolic programs or task graphs that the LLMs can inspect and modify [4], [8]; and (4) skill-level or affordance databases indexed via embeddings for feasibility checking [7], [13]. However, these pipelines often lack causal traceability and spatiotemporal constraints, limiting their ability to resolve ambiguities like spatial relations [6], [23]. In contrast, ***KECRA*** introduces a structured knowledge graph that captures prior operator interactions, encoding instructions, sub-task sequences, environment context and correction paths. This design enables LLMs to retrieve and reason over causally coherent examples, resolving spatial ambiguities and supporting preference-aware adaptation with higher consistency.

3 Methods

We address two key challenges in language-conditioned cobot control: (1) how to align black-box LLMs with precise, step-by-step cobot execution to ensure transparency and controllability, and (2) how to make the system learn from history experience to human preferences for faster and more accurate task completion. To tackle this, we design **KECRA**, a unified framework that integrates three components: (1) VFMs for real-time semantic scene understanding; (2) a Graph agent for structured reasoning and HIL planning; and (3) an online knowledge graph that stores interaction history as causally consistent subgraphs to support retrieval-augmented reasoning. Together, they form a perception–planning–memory loop for transparent, adaptive cobot manipulation.

3.1 Capturing Environment Context with VFMs

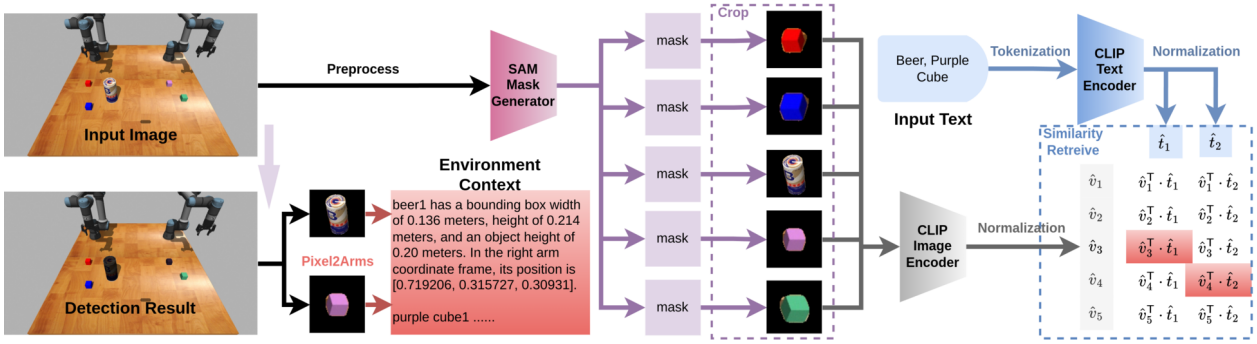

Figure 2: The environment context capturing with example input "Beer, Purple Cube".

During detection (Figure 2), object names are tokenized and encoded with the pretrained *Contrastive Language–Image Pretraining* (CLIP) [24] text encoder to obtain normalized text feature vectors $\hat{\mathbf{t}}_k \in \mathbb{R}^d$. In parallel, pretrained *Segment Anything Model* (SAM) [25] generates segmentation masks from RGB images, which are cropped and encoded by the CLIP image encoder to produce image features $\hat{\mathbf{v}}_i \in \mathbb{R}^d$. Recognition is performed by cosine similarity followed by softmax normalization:

$$Score(i) = \frac{\exp\left(\hat{\mathbf{v}}_i^\top \hat{\mathbf{t}}_k\right)}{\sum_{j=1}^{M} \exp\left(\hat{\mathbf{v}}_j^\top \hat{\mathbf{t}}_k\right)}, \tag{1}$$

where $Score(i)$ is the probability that candidate i belongs to class k. A match is accepted if $Score(i) > 0.2$, a threshold selected from validation to ensure reliable discrimination.

For each matched object, its pixel center (x_c, y_c) is derived from the bounding box, and height is estimated from the depth value at (x_c, y_c). Using camera intrinsics and depth, 2D coordinates are projected into 3D positions and dimensions, then transformed into each arm's coordinate frame by extrinsic calibration. Finally, this context is structured into a natural language environment description, as shown in Figure 2.

3.2 Planning with Graph Agent

KECRA's planning module employs a state graph agent built with LangGraph [26], enabling structured reasoning with human oversight. Unlike linear prompt chains, the agent uses explicit Chain-of-Thought (CoT) decomposition to break high-level instructions into verifiable steps, improving interpretability and reducing hallucinations. HIL checkpoints allow operators to inspect and revise plans at each stage for safe mid-execution correction.

The planner consists of seven nodes: *Start*, *Plan Tasks*, *Confirm Tasks*, *Generate Poses*, *Confirm Poses*, *End*, and the error-handling node *Pose Fix* (Figure 3). At *Plan Tasks*, the LLM integrates the instruction with environmental context and similar past episodes, producing a sub-task list ($tasks$). This list is verified at *Confirm Tasks*; if rejected, operator feedback ($feedback$) triggers iterative replanning.

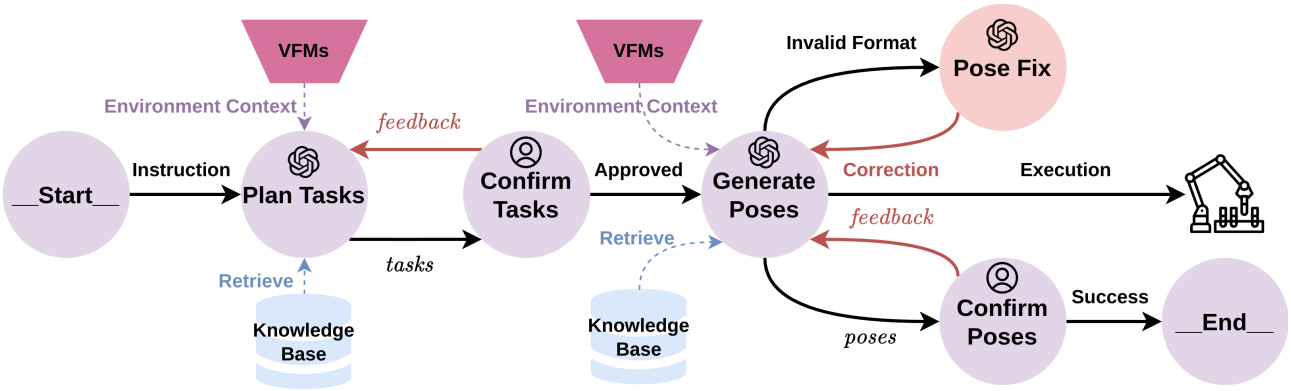

Figure 3: KECRA's state graph planning workflow.

Once approved, *Generate Poses* converts $tasks$ into pose sequences $[x, y, z, roll, pitch, yaw, arm, gripper]$, stored as $poses$. Invalid outputs are redirected to *Pose Fix* for correction. Accepted $poses$ are streamed to the robot arms and verified at *Confirm Poses*. If execution fails (e.g., object slippage), updated context and retrieved correction episodes guide replanning. The loop continues until poses are successfully executed, after which the graph transitions to *End*.

3.3 Memory Enhancement with Knowledge Graph

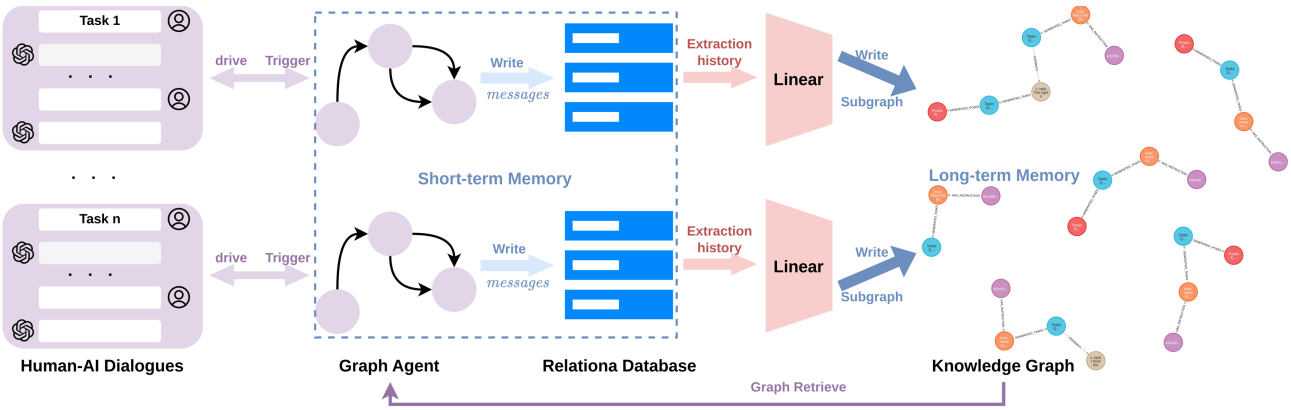

Figure 4: Overview of KECRA's memory architecture.

KECRA's memory system (Figure 4) has two components: (1) **short-term memory**, which stores ongoing interaction context (instructions, task lists, poses, feedback) in the state variable $messages$ and persists it via checkpointing; and (2) **long-term memory**, which transforms completed task histories into a knowledge graph in a graph database, linking instructions, sub-tasks, poses, and feedback.

When the state graph reaches *End*, each episode is unfolded into a causal subgraph (e.g., *Instruction* $\rightarrow$ *TaskList* $\rightarrow$ *PoseList* with feedback links) and added to the knowledge graph. This graph supports RAG through three mechanisms: (1) **Plan Tasks**, where the agent retrieves instructions similar to the operator's input and uses the final task and pose lists of mature episodes (with few corrections) as few-shot references; (2) **Confirm Tasks**, where task revisions trigger a search for past episodes with similar corrections, and the before/after task spans are supplied as guidance for replanning; and (3)

Confirm Poses, where execution failures prompt retrieval of prior pose-level corrections, using the transition from incorrect to corrected poses as regeneration examples.

4 Experiment

4.1 Experiment Setup

Simulation Environment: Experiments are conducted in Gazebo with two UR5e arms and four colored cubes. Two cubes are reachable only by the left arm and two only by the right, creating scenarios for both single-arm and coordinated dual-arm manipulation.

Language Models: KECRA uses GPT-4o for reasoning and *text-embedding-3-small* [27] for similarity retrieval in RAG, both via the OpenAI API.

Task Categories: We evaluate three categories: (1) single-arm placement, (2) cross-arm handover and placement, and (3) long-horizon instructions. Each includes 100 trials, with five linguistic variants per instruction to test robustness.

Baselines and Ablations: We compare KECRA to: (1) a prompt-only baseline using GPT-4o, where the prompt is optimized with demonstrations extracted from KECRA's own successful dialogues to improve fairness, but without feedback or memory mechanisms, and (2) an ablation without the knowledge graph, isolating its contribution.

Metrics: Performance is measured by **task success rate** and **average execution steps**. The prompt-only baseline completes in one step, while KECRA requires at least four (*Plan* $\rightarrow$ *Confirm Tasks* $\rightarrow$ *Generate Poses* $\rightarrow$ *Confirm Poses*); each additional human feedback adds two steps.

4.2 Experiment Results

From Table 1 we observe KECRA-full delivers the highest reliability, achieving 95%, 89%, and 77% success on the single-arm, cross-arm and long-horizon sets, respectively. The checkpointed state graph restricts the execution flow to at least four steps, while feedback loops raise the average to 4.38–6.74, but they prevent most infeasible plans.

Removing the knowledge graph lowers success by 9 percentage points on average and increases the step cost to 6.78–10.13. Without retrieval, the agent requires additional feedback rounds to eliminate hallucinated sub-tasks or kinematically invalid poses, confirming the benefit of experience reuse.

The prompt-only baseline succeeds in at most 71% of single-arm cases and almost never solves long-horizon tasks; its single step planning precludes intermediate correction. For example, in a long-horizon scenario ("stack the green cube on the blue, then move it to the left of the purple cube"), the baseline often failed to execute the second step because the initial plan ignored the change in object location after the first action, leading to incorrect grasp positions. KECRA succeeded by incorporating mid-execution corrections and retrieving prior episodes, enabling it to adjust the plan and trajectory accordingly. These results demonstrate that: (1) a transparent loop state graph with human checkpoints markedly improves task completion, and (2) the automatically accumulated knowledge graph accelerates convergence by supplying specific examples for retrieval-augmented reasoning.

Table 1: Success rate (%) and average execution steps.

	Success ↑			Steps ↓		
Task	Single	Cross	Long	Single	Cross	Long
KECRA (full)	**95**	**89**	**77**	**4.38**	**4.99**	**6.74**
KECRA – w/o KG	86	78	55	6.78	7.19	10.13
Prompt-only baseline	71	28	5	1.00	1.00	1.00

5 Conclusion

We develop and validate KECRA as a transparent controller for cobot operations. It transforms black-box LLMs into transparent cobot controllers through synergistic integration of visual perception (VFMs), a rewindable state-graph agent for human verified planning, and an automatically constructed knowledge graph capturing causal interaction histories. Validation in dual-arm tabletop tasks demonstrates that KECRA significantly outperforms prompt-only baseline. In complex cross-arm and long-horizon scenarios, KECRA maintains high reliability while substantially reducing correction cycles.

References

[1] T. Wang, C. Mao, B. Sun, and Z. Li, “Genealogy of construction robotics”, *Autom. Constr.*, vol. 166, p. 105 607, 2024.

[2] M. Schöberl, A. Huber, S. Kreppold, J. Dirnaichner, S. Kessler, and J. Fottner, “Cobot uptake in construction: Embedding collaborative robots in digital construction processes”, *Constr. Robot.*, vol. 7, no. 1, pp. 89–103, 2023.

[3] H. Zhou et al., *Bridging language and action: A survey of language-conditioned robot manipulation*, arXiv preprint arXiv:2312.10807, 2025.

[4] J. Liang et al., “Code as policies: Language model programs for embodied control”, in *Proc. of the ICRA*, 2023, pp. 9493–9500.

[5] S. Huang, Z. Jiang, H. Dong, Y. Qiao, P. Gao, and H. Li, *Instruct2act: Mapping multi-modality instructions to robotic actions with large language model*, arXiv preprint arXiv:2305.11176, 2023.

[6] L. Zha et al., “Distilling and retrieving generalizable knowledge for robot manipulation via language corrections”, in *Proc. of the ICRA*, 2024, pp. 15 172–15 179.

[7] M. Ahn et al., *Do as i can, not as i say: Grounding language in robotic affordances*, arXiv preprint arXiv:2204.01691, 2022.

[8] I. Singh et al., “Progprompt: Generating situated robot task plans using large language models”, in *Proc. of the ICRA*, 2023, pp. 11 523–11 530.

[9] M. Zhang, R. Xu, H. Wu, J. Pan, and X. Luo, “Human–robot collaboration for on-site construction”, *Autom. Constr.*, vol. 150, p. 104 812, 2023.

[10] OpenAI, *Gpt-4o system card*, arXiv preprint arXiv:2410.21276, 2024.

[11] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, “Language models as zero-shot planners: Extracting actionable knowledge for embodied agents”, in *Proc. of the ICML*, PMLR, 2022, pp. 9118–9147.

[12] W. Huang et al., “Inner monologue: Embodied reasoning through planning with language models”, in *Proc. of the CoRL*, 2022.

[13] A. Z. Ren et al., “Robots that ask for help: Uncertainty alignment for large language model planners”, in *Proc. of the CoRL*, 2023.

[14] W. Yu et al., “Language to rewards for robotic skill synthesis”, in *Proc. of the CoRL*, 2023.

[15] S. Mirchandani et al., “Large language models as general pattern machines”, in *Proc. of the CoRL*, 2023.

[16] C. Rivera et al., *Conceptagent: Llm-driven precondition grounding and tree search for robust task planning and execution*, arXiv preprint arXiv:2410.06108, 2024.

[17] S. Izadi and M. Forouzanfar, “Error correction and adaptation in conversational ai: A review of techniques and applications in chatbots”, *Ai*, vol. 5, no. 2, pp. 803–841, 2024.

[18] Z. Liu, A. Bahety, and S. Song, “REFLECT: Summarizing robot experiences for failure explanation and correction”, in *Proc. of the CoRL*, 2023.

[19] P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive nlp tasks”, in *Proc. of the NeurIPS*, 2020, pp. 9459–9474.

[20] Z. Li, Z. Wang, W. Wang, K. Hung, H. Xie, and F. L. Wang, “Retrieval-augmented generation for educational application: A systematic survey”, *Comput. Educ.: Artif. Intell.*, vol. 8, p. 100 417, 2025.

[21] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, “Llm-planner: Few-shot grounded planning for embodied agents with large language models”, in *Proc. of the ICCV*, 2023, pp. 2998–3009.

[22] S.-Q. Yan, J.-C. Gu, Y. Zhu, and Z.-H. Ling, *Corrective retrieval augmented generation*, arXiv preprint arXiv:2401.15884, 2024.

[23] Y. Yuan et al., *From seeing to doing: Bridging reasoning and decision for robotic manipulation*, arXiv preprint arXiv:2505.08548, 2025.

[24] A. Radford et al., “Learning transferable visual models from natural language supervision”, in *Proc. of ICML*, PmLR, 2021, pp. 8748–8763.

[25] A. Kirillov et al., “Segment anything”, in *Proc. of the ICCV*, 2023, pp. 4015–4026.

[26] J. Wang and Z. Duan, *Agent ai with langgraph: A modular framework for enhancing machine translation using large language models*, arXiv preprint arXiv:2412.03801, 2024.

[27] OpenAI, *Openai embeddings api*, 2024. [Online]. Available: https://platform.openai.com/docs/guides/embeddings