Proceedings of the 5th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/116/2025.20428

Strategies of Building AI Agents for Multimodal Productivity with Contemporary Large Languag Models

Xiaotian Li[1,a,*]

[1]*Faculty for Natural Sciences, Norwegian University of Science and Technology, Trondheim, Norway*
a. invictuate@gmail.com
**corresponding author*

Abstract: The past few years have witnessed significant advancements in generative artificial intelligence (AI) led by large language models (LLMs), applications demonstrating capabilities in traditionally unattainable tasks. Numerous efforts are being initiated exploring a prospect all the more exciting, to employ LLMs not just as language processors, but as a starting point toward AI agents that can adapt to diverse tasks and complex scenarios. In this paper, a survey is offered on the state-of-the-art strategies of deploying such models for the generation of both text-based domain-specific contents and multimodal outputs embodied in interactions with web applications, industrial software and ultimately the physical world. Two approaches to implementing multimodality are delineated: Direct embedding of multimodal data, and conversion of multimodal data to text. Both types have seen extensive use in areas of research focus, such as image processing, embodied action and software automation. Representative cases from these categories are reviewed with a focus on their input/output modalities, methods of processing multimodal data and output quality.

Keywords: Artificial Intelligence, Computation and Language, Large Language Models, AI Agents.

1. Introduction

AI agents are autonomous instances of applied AI that can perceive, reason and take action on their surroundings [1, 2]. This long-present concept is regarded as a stepping stone in the pursuit of artificial general intelligence; however, early efforts in this development could mostly focus on specific tasks, being constructed on less generalizable symbolic logic or reinforcement learning. In such cases, improvements are restricted to algorithms and training strategies, not adaptability to more diverse scenarios; different applications often require training of different models from scratch.

Promising emergent capabilities have recently been demonstrated by advancements in transformer-based LLM. These AI models are grounded on natural language processing but have been employed as a powerful general task solver model to process various real-world scenarios, such as software development and scientific research.

LLMs are large-scale neural network models trained for natural language processing, specifically to generate human-like text, on broad data and large numbers (to hundreds of billions) of parameters [3]. Most contemporary LLMs follow a transformer-based and decoder-only architecture. They inherited distinguishing features of their predecessors, Pre-trained language models (PLMs) and

DOI: 10.54254/2755-2721/116/2025.20428

neural language models (NLMs), being task-agnostic and applicable to unlabeled text corpora, but achieving far stronger language and context understanding abilities. Most notable examples include Pathways Language Model (PaLM), Large Language Model Meta AI (LLaMA), and Generative Pre-trained Transformer (GPT)-4. Agents that adopt LLM as their core decision controller can be guided through complex reasoning and planning tasks through problem decomposition techniques such as Chain-of-Thought (CoT), with performances comparable to individual symbolic agents. They are also able to learn from real-time, in-context feedback during the exploration of various tasks in prompting modes like few-shot and zero-shot generalization. Their primary input/output format in natural language facilitates communication both between agents and between agents and humans, eliminating most needs for code-switching and giving rise to seamless collaboration. For these and many other reasons, LLMs are coming to be considered the most promising foundation for general-purpose AI agents.

LLMs are suitable for extension towards input and output formats other than text [4]. Belonging to the class of foundation models, the mode of action of LLMs requires little alteration to be adapted to equivalently tokenized and embedded data representing images, audio, or video. This concept that an AI entity is able to process different data modes is referred to as multimodality. Many applications of AI are impossible without multimodality, such as those in industries that deal with a mixture of data types, such as robotics, experimental sciences, manufacturing and gaming [5].

Multimodality in LLM-based agents can be implemented either directly or indirectly. The model can have built-in modules for generating embeddings directly from different data formats, concatenating and aligning embedded data in multimodal sentences. Alignment can be handled either at the embedding layers or the attention layer. Visual data, for example, can be embedded using visual transformers to be processed directly in a LLM. An increasing number of published LLMs, such as GPT-4o, have integrated functionalities to process visual data, and have readily served as the basis for this class of agents. In the indirect approach, the LLM core still only processes text as input and output, while other specialized tools, often non-LLM, are responsible for converting other forms of data to and from text. This usage leverages the reasoning capabilities of LLMs while reducing the need for additional training to a minimum.

This paper gives an overview of some representative LLM agents employing each of these strategies in several fields of especially high research interest, such as image processing, embodied action and software/hardware automation. The aim of this survey is to highlight the current capabilities of LLM agents and provide a summary of the alternative approaches toward the same set of most important agent functionalities.

2. Strategies toward multimodality for LLM-based agents

2.1. Multimodality through external tools

In this architecture, the LLM module only accepts text as input. Other input formats are preprocessed into text before further interpretation.

2.1.1. Text to industrial procedures

ChemCrow is a tool-integrated development upon GPT-4 that plans and executes procedures for the synthesis of chemicals [6]. Users give a brief description of the target use of the chemical as its initial prompt. The agent analyzes the request and searches for available options in chemical databases, such as RXN4Chemistry from International Business Machines (IBM) Research, Reaxys, and SciFinder. The completed plan is then sent to RoboRXN, a robotic synthesis platform, to perform experiments on real chemicals. Intermediate results from both the searches and experiment outcomes are fed back to the model for stepwise reasoning. Usage of the tools is guided by human-made documentation,

DOI: 10.54254/2755-2721/116/2025.20428

with minimal training phase otherwise. The agent succeeded in synthesizing 3 different chemical compounds, each taking from 1 to 3 steps. It is also able to assess risks and hazards associated with chemicals.

Coscientist is a model developed for similar purposes by Boiko et al. [7]. At the point of proposal, it retrieves data for chemical reactions primarily from a generic web search instead of specialized chemistry database APIs, but the possibility of integrating such databases has been mentioned. The generated experimental plan can be coded for robotic experimentation platforms such as Emerald Cloud Lab (ECL).

Zhao et al. propose an application of LLM for managing shopfloor logistics on manufacturing sites. It can optimize manufacturing workflow involving different sets of machinery in real time, allocating tasks between more and less occupied units [8]. The agent is composed of separate Bidder and Bid Inviter modules, as well as Thinking and Decision modules. The former collects primary data on the current workload at each unit including tasks in queue, completion rate, or expected completion time, and processes the data into a more high-level formatted documentation. This is then input to the latter modules to generate a schedule on how to allocate subsequent tasks to the units. The generated plans are found to reduce the total amount of time to process all tasks by 20 to 50 percent compared to random.

2.1.2. Text to images

GenArtist is a multimodal agent that can generate images from text prompts as well as edit existing images. Image processing tasks are mostly implemented by utilizing an external tool library, among which image input is handled by StableDiffusion and other editing functionalities their own respective tools such as Stable Diffusion XL (SDXL) and AnyDoor [9]. The LLM module is responsible for breaking down the text prompts into individual features to include in the images and processing them into formats required by different editing tools. Compared to baseline image generation tools, this allows the agent to produce images in greater detail.

2.1.3. Text to software GUI procedures

MindAgent is designed to generate action plans in gaming interactions for Minecraft-like games [10]. In the demonstration, the agent acts on the jointly developed benchmark game CuisineWorld which has a text-based interface. The model supports textual input formats for game scenarios, such as game state descriptions and environmental feedback. It has a defined action space with operations including goto, get and put (item). Training is through in-context learning and chain-of-thought prompting.

2.2. Multimodality at direct input

In this mode, inputs other than text, such as images and video, are directly sent to the LLM embedded by specialized modules.

2.2.1. Text and vision to robotic procedures

PaLM-E is an agent for object manipulation built upon embodiment ("-E") of the pre-trained LLM, PaLM. The input to this agent is vision-language multimodal, where a sentence can be a mixture of text, images and real-time visual input with a vision transformer model [11]. The visual inputs are separated into distinct objects, before being tokenized in a single process as a multimodal sentence. The training dataset is likewise a mixture of text and continuous observations. The reasoning is carried out by decoding a text description of the task and mapping it to robot instructions. The model completes object manipulation tasks in the form of "bring rice chips from the drawer" or "bring the

DOI: 10.54254/2755-2721/116/2025.20428

blue block under the yellow block" at a success rate above 90% and can react to adversarial events such as interim human actions on the objects.

ShapeLLM is developed from LLaMA for recognizing 3D images for multimodal applications. The images are processed by an encoder ReCon++ into a 3D point cloud format for input [12]. After receiving the input, the model can discuss various aspects of the image in text. The training data is focused on instruction-following tuning. The model can be tuned to output low-level instructions for execution by robots, such as describing motion in terms of coordinates, but no specific integration attempts are discussed.

2.2.2. Text and vision to images

SmartEdit is an image generation and editing model utilizing direct visual encoding of input images based on LLaVa with fine tuning by low-rank adaptation [13]. Features from the image are interpreted alongside text prompts through a Bidirectional Interaction Module (BIM) to enhance understanding and reasoning. The visual model is trained with segmentation data for perception of spatial attributes and image context, capable of handling difficult features in images such as mirrors. The agent outperforms several other models e.g. InstructPix2Pix and MagicBrush in handling complex reasoning and achieves good results overall in image quality metrics such as Peak signal-to-noise ratio (PSNR) and Structural Similarity Index (SSIM).

Vision-LLM is a generalist multimodal agent that integrates visual input, reasoning and generation within an end-to-end framework for image generation, analysis and editing [14]. Image inputs are treated by the vision foundation model Contrastive Language-Image Pre-Training (CLIP), and visual prompts are region-embedded through the use of binary masks followed by processing through convolutional layers. For a given set of input images and text prompts, the agent gives a text summary of the image that also contains suggested options for further actions in code words, including [DET] for detection, [SEG] for segmentation and [GEN] for generation. These, termed routing tokens, can mark the calling of subsequent decoder modules for the appropriate task. Training of the model is comprised of multimodal training, multi-capacity and decoder-only fine tuning. Its capability in object recognition and segmentation is above other generalist models and comparable with specialist models, and image output quality is above baseline StableDiffusion.

2.2.3. Text and vision to software GUI procedures

AppAgent is an LLM agent described by Zhang et al. trained for the generalized operation of smartphone applications, based on GPT-4V with built-in image processing capabilities [15]. It takes text prompts as input alongside documentation of the apps' functionalities and dynamically makes step-by-step decisions by recognizing screenshots of the app UI during the procedure. During use, the agent first generates a text description of the next step and then chooses the relevant action from a set list of available options, such as tap or text. The agent achieved an overall 73.3% success rate in tasks shorter than 10 steps. This includes visual tasks such as image editing, where the agent adjusts the contrast and exposure of photos at its own discretion.

MobileFlow is a multimodal LLM employing a Qwen-7B-based language model that generates procedures for graphic user interface (GUI) interaction tasks from text prompts and UI images [16]. Its visual perception leverages a vision transformer OpenCLIP and. UI image interpretation is trained with human-labeled image datasets for visual-language alignment. Its action space is defined in the prompt as tap, scroll and text. The LLM adopts a Mixture of Experts (MoE) expansion using trained multilayer perceptron as initial layers. In deployment, it is prompted in a chain-of-thought manner to analyze its observations of an UI state and plan its next steps given a task and historical actions. Tasks such as food ordering, insurance and medical arrangements as well as gaming are attempted,

DOI: 10.54254/2755-2721/116/2025.20428

categorized by the number of expected steps. Success rates are on average 30 to 50% higher than baseline vision-capable LLMs.

3. Limitations of multimodal LLM agents

Multimodal LLM agents are susceptible to errors common to LLM interpretation and reasoning processes, the major class of which are collectively known as hallucination. This can be amplified in multimodal contexts since cross-checking different input sources can be more difficult. Zhong et al. in an analysis of an image interpretation model found that errors initially made while analyzing the image are unlikely to be corrected in subsequent text interactions without intervention, leading to what is referred to as hallucination snowballing [17]. For example, if the model incorrectly recognizes a red traffic light as green, all its further outputs will be based on this assumption by default even if it could correct itself if separately asked. Methods for mitigating this include revising the subsequent textual outputs with the one derived from visual input and providing models with back reference to the initial visual information.

Yu et al. assessed the performances and potential sources of error in LLM agents for chemistry during general reasoning and use of tools [18]. It is found that reasoning error predominates when the tasks provided are more generalized, and cannot be reliably avoided even when correct data has been retrieved from an external tool. For example, the model can correctly calculate the required reaction ratio of two chemicals by invoking an external calculator but fails to compare it to the provided quantities in an attempt to decide whether one of the two is in excess. For more specific tasks that require less reasoning, such as converting chemical formulae and names, reasoning error is found to be negligible compared to tool output error.

4. Conclusion

This paper presents a treatment of various recent exemplars in achieving and applying multimodality in LLM-based AI agents, overviewing their modes of action. Discussion focuses on the agents' intended input and output formats, ways of processing non-text data, training needs, and prompting and reasoning modes. Most such agents reviewed follow a similar principle at the core of the reasoning process, by prompting the agent to vocalize its intermediate reasoning results and subsequently narrow down a range of action options. This process maximizes the LLM's textual reasoning potential, with which the agents demonstrate promising potential over various task types and benchmarks. Overall, LLM-based agents can better interpret contexts and personal preferences and are efficient in mapping complex reasoning onto a small set of operations. This most notably stems from their capability to dynamically perceive updated input during the full decision-making process, using them as new context to adjust subsequent actions. Weaknesses common in LLM reasoning processes are still present, but as in language tasks, they can be controlled by limiting over-general prompts, avoiding core numerical reasoning, and including mechanisms to review and validate information.

References

[1] *Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., Zhou, E., Zheng, R., Fan, X., Wang, X., Xiong, L., Zhou, Y., Wang, W., Jiang, C., Zou, Y., Liu, X., Yin, Z., Dou, S., Weng, R., Cheng, W., Zhang, Q., Qin, W., Zheng, Y., Qiu, X., Huang, X., Gui, T. (2023). The Rise and Potential of Large Language Model Based Agents: A Survey. arXiv preprint arXiv:2309.07864.*

[2] *Liu, J., Wang, K., Chen, Y., Peng, X., Chen, Z., Zhang, L., Lou, Y. (2024). Large Language Model-Based Agents for Software Engineering: A Survey. arXiv preprint arXiv:2409.02977.*

[3] *Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., Gao, J. (2024). Large Language Models: A Survey. arXiv preprint arXiv:2402.06196.*

DOI: 10.54254/2755-2721/116/2025.20428

[4] Huyen, C. (2023, October 10). Multimodality and Large Multimodal Models (LMMs). Chip Huyen. https://huyenchip.com/2023/10/10/multimodal.html

[5] Raschka, S., PhD. (2024, November 3). Understanding Multimodal LLMs. Ahead of AI. https://magazine.sebastianraschka.com/p/understanding-multimodal-llms

[6] Bran, A.M., Cox, S., Schilter, O., Baldassari, C., White, A.D., Schwaller, P. (2023). ChemCrow: Augmenting large-language models with chemistry tools. arXiv preprint arXiv:2304.05376.

[7] Boiko, D. A., MacKnight, R., Kline, B., & Gomes, G. (2023). Autonomous chemical research with large language models. Nature, 624(7992), 570–578. https://doi.org/10.1038/s41586-023-06792-0

[8] Zhao, Z., Tang, D., Zhu, H., Zhang, Z., Chen, K., Liu, C., Ji, Y. (2024). A Large Language Model-based multi-agent manufacturing system for intelligent shopfloor. arXiv preprint arXiv:2405.16887.

[9] Wang, Z., Li, A., Li, Z., Liu, X. (2024). GenArtist: Multimodal LLM as an Agent for Unified Image Generation and Editing. arXiv preprint arXiv:2407.05600.

[10] Gong, R., Huang, Q., Ma, X., Vo, H., Durante, Z., Noda, Y., Zheng, Z., Zhu, S., Terzopoulos, D., Fei-Fei, L., Gao, J. (2023). MindAgent: Emergent Gaming Interaction. arXiv preprint arXiv:2309.09971.

[11] Driess, D., Xia, F., Sajjadi, M.S.M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., Huang, W., Chebotar, Y., Sermanet, P., Duckworth, D., Levine, S., Vanhoucke, V., Hausman, K., Toussaint, M., Greff, K., Zeng, A., Mordatch, I., Florence, P. (2023). PaLM-E: An Embodied Multimodal Language Model. arXiv preprint arXiv:2303.03378.

[12] Qi, Z., Dong, R., Zhang, S., Geng, H., Han, C., Ge, Z., Yi, L., Ma, K. (2024). ShapeLLM: Universal 3D Object Understanding for Embodied Interaction. arXiv preprint arXiv:2402.17766.

[13] Huang, Y., Xie, L., Wang, X., Yuan, Z., Cun, X., Ge, Y., Zhou, J., Dong, C., Huang, R., Zhang, R., & Shan, Y. (2024). SmartEdit: Exploring Complex Instruction-Based Image Editing with Multimodal Large Language Models. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 33, 8362–8371. https://doi.org/10.1109/cvpr52733.2024.00799

[14] Wu, J., Zhong, M., Xing, S., Lai, Z., Liu, Z., Wang, W., Chen, Z., Zhu, X., Lu, L., Lu, T., Luo, P., Qiao, Y., Dai, J. (2024). VisionLLM v2: An End-to-End Generalist Multimodal Large Language Model for Hundreds of Vision-Language Tasks. arXiv preprint arXiv:2406.08394.

[15] Zhang, C., Yang, Z., Liu, J., Han, Y., Chen, X., Huang, Z., Fu, B., Yu, G. (2023). AppAgent: Multimodal Agents as Smartphone Users. arXiv preprint arXiv:2312.13771.

[16] Nong, S., Zhu, J., Wu, R., Jin, J., Shan, S., Huang, X., Xu, W. (2024). MobileFlow: A Multimodal LLM For Mobile GUI Agent. arXiv preprint arXiv:2407.04346.

[17] Zhong, W., Feng, X., Zhao, L., Li, Q., Huang, L., Gu, Y., Ma, W., Xu, Y., Qin, B. (2024). Investigating and Mitigating the Multimodal Hallucination Snowballing in Large Vision-Language Models. arXiv preprint arXiv:2407.00569.

[18] Yu, B., Baker, F.N., Chen, Z., Herb, G., Gou, B., Adu-Ampratwum, D., Ning, X., Sun, H. (2024). Tooling or Not Tooling? The Impact of Tools on Language Agents for Chemistry Problem Solving. arXiv preprint arXiv:2411.07228.