My research applies large language models (LLMs) to the core problem of AI: building autonomous agents to interact with the world. Taking insights from natural language processing (NLP), reinforcement learning (RL), and cognitive science (CogSci), I have made core contributions to establishing the emerging field of "**language agents**" (Fig. 1), which ground LLMs for sequential decision-making in various digital and physical environments, and unlock next-gen AI products like ChatGPT plugins and Microsoft Copilot. Concretely,

1. I have developed **foundational and domain-agnostic methods** to adapt LLM reasoning for acting [26], learning [12], and planning [24], which are widely adopted in various domains, such as art [14], healthcare [7], robotics [6], education [1], disaster control [2], fact checking [10], networks [5], and autonomous driving [3].
2. I have introduced **new types of AI benchmarks** based on scalable and practical language interactions, such as web shopping [17], coding [28], and software engineering [8], which are hard to solve but easy to evaluate and transfer to real-world usage.
3. I have **proposed a theoretical framework** [23] inspired by symbolic AI and CogSci to organize various language agents in simple, unified terms, blueprint their future developments, and formulate their study as an independent and interdisciplinary subject.

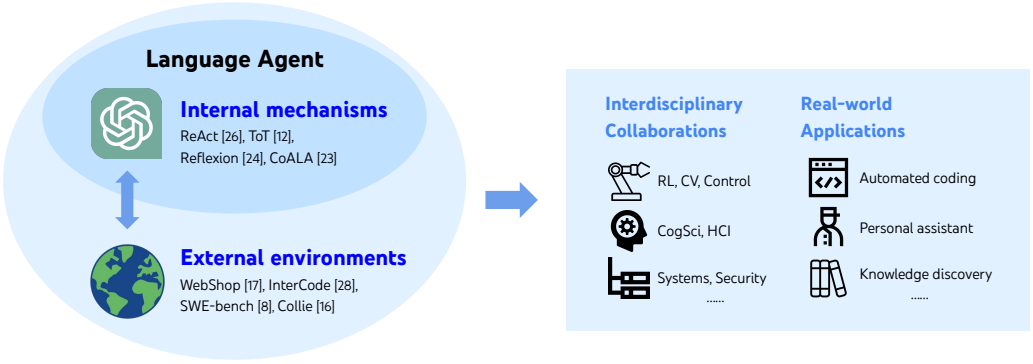

Figure 1: I study language agents, a new kind of autonomous agents based on large language models.

Four years ago, when the potential of language models was just being tapped by GPT-2, and only small-scale combinations of language and RL were explored in individual toy tasks, my work [22] was one of the first to envision and explore agents based on language models, capable of acting in various environments using pre-trained knowledge and minimal adaption. Today, LLMs have undergone significant advancements, and the need for such interactive language agents has become evident: even the biggest LLMs like GPT-4 know limited things and hallucinate. But by generating actions and accessing feedback iteratively, language agents can obtain new knowledge, correct mistakes, and interface with the world to achieve goals. Compared to rule-based or RL agents, they also alleviate intensive, domain-specific heuristics design or training, accessible for non-experts to develop various applications.

In the next few years, I envision language agents being responsibly and widely deployed in digital and physical worlds, automating various tasks in our work and life, and discovering new knowledge in science. **I am excited to gain insights from and collaborate with researchers in various areas**, like systems, security, policy, programming languages (PL), human-computer interaction (HCI), computer vision (CV), NLP, CogSci, RL, robotics, theory, and computational sciences, in order to advance such an interdisciplinary future agenda. Below is a summary of my past work and some more concrete future research directions.

1. Developing Language Agents that Reason to Act

Without actions and feedback, LLMs cannot avoid hallucination or affect the world. But once grounded, the challenge often switches to an explosion of possible actions, due to the compositionality of language and open-endedness of environments. For example, a web agent could type billions of search queries and navigate millions of sites, as opposed to Atari agents with tens of fixed actions. This is exacerbated when only a few example trajectories are given in the LLM context for imitation, as opposed to millions of training steps in RL.

To improve decision-making in large action spaces, my work ReAct [26] proposed the key idea that the action space of a language agent (e.g., `search`[*a query*]) can be augmented by reasoning (`think`[*a thought*]). Correspondingly, humans can easily annotate thoughts that explain and inform their actions, so that agents can leverage the reasoning pattern to better generalize in new scenarios (Fig. 2). ReAct achieved state-of-the-art results across diverse NLP and RL domains, and boosts human-agent alignment: unlike black box RL policies, humans can examine and diagnose ReAct agents via text thoughts, and modify them to correct or control agent behaviors. Due to its simplicity and generality, ReAct has been the most widespread method for language agents, the basis of follow-up projects like AutoGPT, and the core of the LangChain package downloaded by millions of users to develop LLM-based applications.

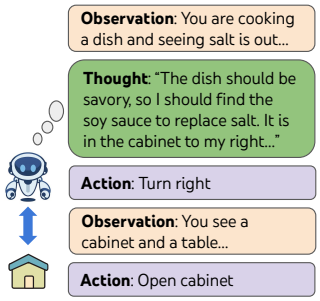

Figure 2: In ReAct [26], reasoning informs and improves acting.

However, LLMs generate reasoning token by token without backtracking or looking ahead, which becomes brittle as mistakes inevitably occur over a long horizon, yet autoregressive inference lacks mechanisms to correct and avoid them. One of my solutions is to introduce feedback signals, like runtime errors or self-created unit test results in coding, to interrupt agents from acting to learning: Reflexion [12] uses LLM reasoning to self-reflect, and store reflections across trials to self-improve via language update instead of gradient descent. Another of my solutions, Tree of Thoughts (ToT) [24], proposes to augment LLMs with a tree search algorithm (e.g., DFS, BFS) that maintains diverse continuation choices of reasoning ("thoughts") and explores them systematically using LLM evaluation as search heuristics (Fig. 3). Using GPT-4 on Game of 24 and Crosswords, ToT improves the popular Chain of Thoughts (CoT) prompting from 7%/1% to 74%/35% respectively, showing the potential of how classical algorithms and LLMs can be synergized.

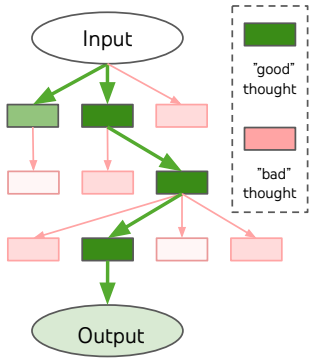

Figure 3: In ToT [24], LLMs systematically generate and evaluate thoughts in a tree search.

2. Benchmarking Language Agents with Digital Applications

Benchmarks for traditional AI agents struggle to be both practical and scalable. Practical tasks that interact with humans (e.g., dialogue) or physical environments (e.g., robotics) face difficulty collecting trajectories or reward signals at scale. Thus most RL algorithms are developed in digital games or simulations with unlimited interactions and rewards, but are hard to transfer to real-world usage. To develop language agents, my work has created a new

kind of AI problem: practical digital applications in scalable language-based environments.

One such domain is the Internet, which has a colossal scale with numerous applications. Despite prior efforts to benchmark web interaction, the key challenge often is **scalable and faithful evaluation**: web trajectories could be long, complex, and diverse, making it hard to compare to a reference trajectory, unreliable to adopt LLM-based evaluation, and expensive to collect human judgments. My key idea is to find a domain that is easy to evaluate the outcome rather than the process. Thus we built WebShop [17], a shopping website environment with millions of real-world products, where an agent needs to read webpages, type queries, and click buttons to buy a product that satisfies the user instruction. It challenges visual understanding, reading comprehension, long-horizon exploration, and provides simple and faithful rewards by comparing the instruction and chosen product attributes. It has been widely used by OpenAI, Google, and other labs for agent evaluation and development.

Another useful and data-rich domain is programming, with unit tests as a desirable outcome evaluation. However, prior benchmarks focused on simple problems (often solvable within ten lines) in a sequence-to-sequence (seq2seq) setup, whereas humans program interactively with execution feedback for much harder problems. This motivated InterCode [28] and SWE-bench [8], where we transformed seq2seq datasets into interactive environments (Fig. 4), and real-world GitHub issues into repository-scale code understanding and generation challenges. These represent two critical directions toward automating real-world software engineering.

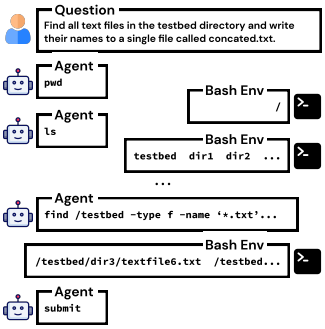

Figure 4: InterCode [28] agents leverage execution feedback in code terminals to better code.

Besides web and code interaction, I have also created practical and scalable tasks to generate text under constraints [16] and answer character-related questions in long books [29] or TV scripts [11], which require agentic interactions to either incorporate feedback or navigate long context and serve to evaluate and develop useful language agents in a sustainable way.

3. Formulating Language Agents with Interdisciplinary Insights

Language agents as a new subject in AI have had a vast array of empirical projects and ideas, yet a lack of standardized terms and unifying frameworks make it hard to compare, organize, or understand different methods that are described in customized ways. Inspired by ideas from computer systems, symbolic AI, and human cognition, I helped propose Cognitive Architectures for Language Agents (CoALA) [23], a simple yet complete framework to express each language agent by (i) the memory modules, (ii) the action space (Fig. 5), and (iii) the decision-making procedure over actions. It helps concretely define terms (e.g., to learn is to write long-term memory), which in turn points out various potential developments (e.g., learning by updating an agent's code or prompt). My research has also intersected with RL and control [22, 20, 27, 9, 4], computer vision [18, 13], computational linguistics [21], multi-agent systems [25], and HCI [14]. These interdisciplinary views and insights will be vital for pursuing the future directions below.

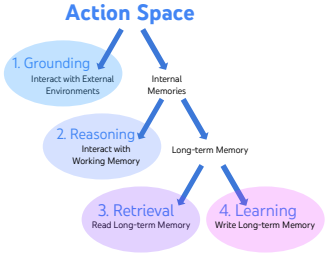

Figure 5: CoALA [23] systematizes the action space of each language agent into four parts.

4. Ongoing and Future Directions

Training LLMs for agents. Most open-source LLMs perform poorly on agent tasks as they were not trained to act, and proprietary models like GPT-4 are expensive to use and lack transparency. My work has shown training LLMs how to reason and use tools leads to a stronger generalization than either alone (Fig. 2). I am excited to work with NLP and systems researchers to develop more effective and efficient open-source LLMs for agents, and establish a reciprocating cycle where better LLMs enable exploration of agent design, and strong agents in turn provide training data to shape LLMs. I also want to work with CV and RL researchers to build agent backbones in multimodal and embodied setups, like a general-purpose computer agent reading screen pixels and using the mouse and keyboard.

Robust and safe deployment. Language agents indicate great opportunities for task automation, personal freedom, and social progress, but also enhanced potential harms like deleting files or attacking servers. I believe it takes concrete and multidisciplinary efforts to better understand and control these emerging systems, such as statistical and mathematical characterization [21] of their capabilities and robustness, defining threat models and finding defenses, and engaging ethics, law, and policy experts in capturing and shaping their societal impact [19]. Across these efforts, it is important to have a holistic view of not just LLMs, but how they are and will be used to interact with the world. CoALA [23] could help organize and guide these efforts, e.g., we could analyze and control risks by defining the action space of language agents (Fig. 5). Another important direction is automated coding [28, 8] (Fig. 4), as agent-generated programs can act in more interpretable and reliable ways than agents.

Knowledge and scientific discovery. So far, the success of LLMs and language agents relies mostly on imitating patterns of how humans write and act, thus happening mostly on tasks that humans have already explored and summarized knowledge about. But to go beyond imitation, we need to equip language agents with intrinsic rewards like curiosity [20], means to planning [24] (Fig. 3) and reinforcement learning [12] using such intrinsic rewards, and a long-term memory [23] to maintain experience, knowledge, and skills. I envision agents that navigate gigantic networks of knowledge (e.g., via ArXiv APIs) to answer self-asked questions, and learn by checking follow-up research via citations [15], interacting with humans [14], or coding [28] (Fig. 4), similar to how Ph.D. students expand human knowledge.

Understanding and helping humans. My work has been inspired by human cognition [24, 23] to build autonomous agents that solve hard tasks with minimal guidance. But to deploy language agents in our society, they will need to infer human intention, invoke and incorporate human feedback, and collaborate with humans or other agents. I hope to engage insights from pragmatics, game theory, social cognition, and HCI to understand how humans perceive language agents [14], and how agents could in turn better model and interact with humans. A particularly exciting domain is education, where I want to develop a tutor agent with a long-term memory [23] of agent-student interaction histories and the student profile to customize education for each student. Beyond teaching existing knowledge , I also envision agents that communicate their discovered concepts (e.g., Move 37 of AlphaGo) to humans by linking their "emergent languages" to human ones [25]. These will help ensure that AI complements and augments human abilities, rather than surpassing or replacing them.

References

[1] Toktam B.Tabrizi, Ozgur Gocer, Arash Sadrieh, and Anastasia Globa. Leveraging ai to instruct architecture students on circular design techniques and life cycle assessment. *9th International Conference on Higher Education Advances (HEAd'23)*, 2023.

[2] Grace Colverd, Paul Darm, Leonard Silverberg, and Noah Kasmanoff. Floodbrain: Flood disaster reporting by web-based retrieval augmented generation with an llm. *ArXiv*, abs/2311.02597, 2023.

[3] Daocheng Fu, Xin Li, Licheng Wen, Min Dou, Pinlong Cai, Botian Shi, and Y. Qiao. Drive like a human: Rethinking autonomous driving with large language models. *ArXiv*, abs/2307.07162, 2023.

[4] Yi Gu, **Shunyu Yao**, Chuang Gan, Joshua B Tenenbaum, and Mo Yu. Revisiting the roles of" text" in text games. In *EMNLP Findings*, 2022.

[5] Pouya Hamadanian, Behnaz Arzani, Sadjad Fouladi, Siva Kesava Reddy Kakarla, Rodrigo Fonseca, Denizcan Billor, Ahmad Cheema, Edet Nkposong, and Ranveer Chandra. A holistic view of ai-driven network incident management. In *ACM Workshop on Hot Topics in Networks*, 2023.

[6] Abdelhadi Hireche, Abdelkader Nasreddine Belkacem, Sadia Jamil, and Chao Chen. Newsgpt: Chatgpt integration for robot-reporter. *ArXiv*, abs/2311.06640, 2023.

[7] Fergus Imrie, Paulius Rauba, and Mihaela van der Schaar. Redefining digital health interfaces with large language models. *ArXiv*, abs/2310.03560, 2023.

[8] Carlos E Jimenez, John Yang, Alexander Wettig, **Shunyu Yao**, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*, 2023.

[9] Yao Mu, **Shunyu Yao**, Mingyu Ding, Ping Luo, and Chuang Gan. EC^2: Emergent communication for embodied control. In *CVPR*, 2023.

[10] Dorian Quelle and Alexandre Bovet. The perils & promises of fact-checking with large language models. *ArXiv*, abs/2310.13549, 2023.

[11] Yisi Sang, Xiangyang Mou, Mo Yu, **Shunyu Yao**, Jing Li, and Jeffrey Stanton. Tvshowguess: Character comprehension in stories as speaker guessing. In *NAACL*, 2022.

[12] Noah Shinn, Beck Cassano, Federico Labash, Ashwin Gopinath, Karthik Narasimhan, and **Shunyu Yao**. Reflexion: Language agents with verbal reinforcement learning. In *NeurIPS*, 2023.

[13] Kevin Smith, Lingjie Mei, **Shunyu Yao**, Jiajun Wu, Elizabeth Spelke, Josh Tenenbaum, and Tomer Ullman. Modeling expectation violation in intuitive physics with coarse probabilistic object representations. In *NeurIPS*, 2019.

[14] Yuqian Sun, Xingyu Li, Ze Gao, Ze Gao, **Shunyu Yao**, Jun Peng, Noura Howell, Tristan Braud, Chang Hee Lee, and Ali Asadipour. ORIBA: Supporting artistic development of original characters with conversational ai agents. *In Submission to CHI*, 2023.

[15] Michael Tang, **Shunyu Yao**, John Yang, and Karthik Narasimhan. Referral augmentation for zero-shot information retrieval. *arXiv preprint arXiv:2305.15098*, 2023.

[16] **Shunyu Yao**, Howard Chen, Austin W Hanjie, Runzhe Yang, and Karthik Narasimhan. Collie: Systematic construction of constrained text generation tasks. *arXiv preprint arXiv:2307.08689*, 2023.

[17] **Shunyu Yao**, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. In *NeurIPS*, 2022.

[18] **Shunyu Yao**, Tzu Ming Harry Hsu, Jun-Yan Zhu, Jiajun Wu, Antonio Torralba, William T Freeman, and Joshua B Tenenbaum. 3d-aware scene manipulation via inverse graphics. In *NeurIPS*, 2018.

[19] **Shunyu Yao** and Karthik Narasimhan. Language agents in the digital world: Opportunities and risks. *princeton-nlp.github.io*, Jul 2023.

[20] **Shunyu Yao**, Karthik Narasimhan, and Matthew Hausknecht. Reading and acting while blindfolded: The need for semantics in text game agents. In *NAACL*, 2021.

[21] **Shunyu Yao**, Binghui Peng, Christos Papadimitriou, and Karthik Narasimhan. Self-attention networks can process bounded hierarchical languages. In *ACL*, 2021.

[22] **Shunyu Yao**, Rohan Rao, Matthew Hausknecht, and Karthik Narasimhan. Keep calm and explore: Language models for action generation in text-based games. In *EMNLP*, 2020.

[23] **Shunyu Yao**, Theodore Sumers, Karthik Narasimhan, and Thomas L Griffiths. Cognitive architectures for language agents. *arXiv preprint arXiv:2309.02427*, 2023.

[24] **Shunyu Yao**, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *NeurIPS*, 2023.

[25] **Shunyu Yao**, Mo Yu, Yang Zhang, Karthik R Narasimhan, Joshua B Tenenbaum, and Chuang Gan. Linking emergent and natural languages via corpus transfer. In *ICLR*, 2022.

[26] **Shunyu Yao**, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *ICLR*, 2023.

[27] Jens Tuyls, **Shunyu Yao**, Sham Kakade, and Karthik Narasimhan. Multi-stage episodic control for strategic exploration in text games. In *ICLR*, 2022.

[28] John Yang, Akshara Prabhakar, Karthik Narasimhan, and **Shunyu Yao**. Intercode: Standardizing and benchmarking interactive coding with execution feedback. In *NeurIPS Datasets and Benchmarks Track*, 2023.

[29] Mo Yu, Jiangnan Li, **Shunyu Yao**, Wenjie Pang, Xiaochen Zhou, Zhou Xiao, Fandong Meng, and Jie Zhou. Personality understanding of fictional characters during book reading. In *ACL*, 2023.