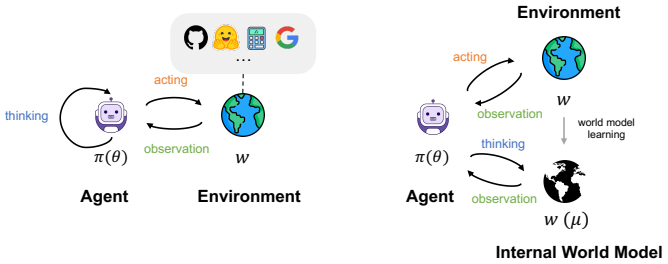
Large language models are rapidly evolving from passive text generators into intelligent agents capable of thinking, acting (e.g., tool use), and long-horizon interaction with the world, as shown in Fig 1(a). Yet current agentic systems still fail in surprisingly fundamental ways. Consider a seemingly simple question: "Who is the current CEO of NVIDIA, and what is the stock price today?" A static LLM may confidently answer with outdated or hallucinated information because it relies entirely on internal parametric memory. In contrast, a tool-augmented agent may issue a dozen redundant web searches for a question that requires only one. These systems fail in opposite directions, but the underlying problem is the same: **the agent does not know when to think internally, when to act externally, and how much computational effort either process deserves.**

This tension between thinking and acting becomes increasingly critical as intelligent agents begin to play central roles in scientific discovery, software engineering, web automation, and everyday decision-making. Modern agents are expected not only to think correctly, but also to interact efficiently with tools, humans, and dynamic environments over sustained horizons. However, despite the rapid progress of agentic AI, existing research remains highly fragmented: reasoning, tool use, safety, evaluation, and alignment are often studied in isolation, without a unified framework for understanding how intelligent behavior arises from the interplay between internal cognition and external action.



(a) ReAct-based Agent (b) Theory of Agent

Figure 1: Demonstration of (a) a ReAct-based agent built upon a thinking–acting–observation loop, and (b) the perspective of Theory of Agent.

In light of these challenges, my research vision is to **lay the groundwork for intelligent agents that think and act in autonomous, controllable, and trustworthy ways.** In pursuit of this vision, my research is built on two coupled pillars: (1) **a principled theoretical framework** — the Theory of Agent (ToA)[1] — that reframes thinking and acting as alternative knowledge-acquisition mechanisms governed by epistemic effort as shown in Fig 1(b); and (2) **a closed-loop methodology, including several practical agentic systems and algorithms**, that operationalizes the theory across training, behavioral control, and evaluation. The two pillars are mutually reinforcing and mutually inspiring: the theory diagnoses failure modes that drive methodological innovation, while methodological discoveries refine and extend the theory. Together, these efforts aim to advance a scientific understanding of intelligent agents and enable the next generation of lifelong agentic systems for real-world deployment.

Theory: A Principled Foundation for Intelligent Agents

My theoretical work is driven by one question: what unifies the seemingly disparate behaviors — reasoning, tool use, planning, interaction — that today's language agents perform? I propose the Theory of Agent (ToA), which views agent intelligence as an adaptive allocation problem: agents must continuously decide how to distribute computation (e.g., tokens) between internal thinking (e.g., planning, reflection, memory) and external interaction (e.g., tool use, communication, execution) under uncertainty and environmental feedback.

Three concepts anchor ToA. First, the knowledge boundary characterizes the limits of an agent's internal competence — what it can answer purely from parameters based on current context (i.e., internal task set $Q_{int}(m, \mathcal{W})$) versus what must require external interactions (i.e., external task set $Q_{ext}(m, \mathcal{W})$). Second, epistemic necessity provides a normative criterion: external tool use is justified if and only if the agent's epistemic uncertainty cannot be reduced through internal thinking alone.

Third, epistemic effort $E(q,m) = E_{int}(q,m) + E_{ext}(q,m)$ decomposes the total informational burden of task q for agent m into an internal component (thinking) and an external component (acting). Critically, any successful trajectory satisfies $E^{\pi}_{int}(q,m) + E^{\pi}_{ext}(q,m) \geq E^{*}(q,m)$ — the minimum epistemic effort cannot be eliminated, only allocated between internal thinking and external acting.

Fig. 2 maps every agent policy to a point in (internal thinking, external acting) space, with the feasibility frontier $E_{int}+E_{ext}=E^{*}$. Specifically, point A corresponds to pure delegation, where the agent relies entirely on external tools to solve the task, while point C represents pure internal reasoning, where the agent attempts to solve the problem without external interaction. Point B denotes the ideal frontier: the minimum amount of external interaction required when internal reasoning is maximally utilized for external tasks. The segments between A, B, and C characterize different optimal trade-offs under varying objectives such as safety, latency, or personalization. The slope β denotes the cost ratio of transition between internal thinking and external acting.

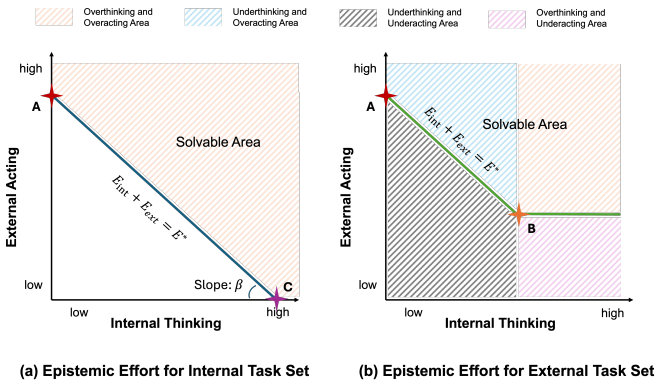


Figure 2: A high-level illustration of epistemic effort allocation and failure modes in tool-augmented agents for: (a) task in $Q_{int}(m,\mathcal{W})$; and (b) task in $Q_{ext}(m,\mathcal{W})$. Figure is adapted from Wang et al. [1].

This framework provides a unified explanation for several canonical failure modes observed in modern language agents — underthinking (e.g., premature termination), overthinking (e.g., 13 redundant solutions to the simple question: what is sum of 2+3?), underacting (e.g., failing to invoke correct tools), and overacting (e.g., 10× redundant steps vs. humans on OSWorld). In all cases, the underlying issue can be viewed as a misallocation of epistemic effort between cognition and action. More importantly, Theory of Agent (ToA) goes beyond descriptive analysis by providing an analytical foundation and actionable guidelines for agent training, regulation, and evaluation.

Method: From Theory to Practical Algorithms and Systems

Building on this unified theory, my work is organized around a closed loop spanning three interconnected thrusts as shown in Fig 3(a): 1) building autonomous agents with strong thinking and acting capabilities for complex real-world environments; 2) making such agents controllable and stable by mitigating fundamental failure modes such as overthinking, overacting and decision oscillation; and 3) ultimately enabling trustworthy agents through multi-dimensional evaluation frameworks that capture realistic deployment constraints, personalization, and safety risks. Together, these efforts aim to advance agent research from fragmented empirical practices toward a practical recipe to build autonomous, controllable and trustworthy agentic systems.

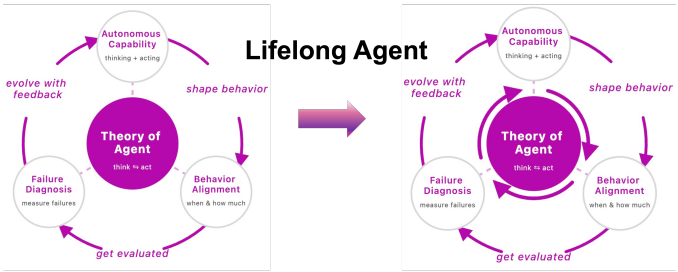


Figure 3: (a) Left: my current research framework for building, regulating, and diagnosing language agents; and (b) Right: my future vision for lifelong agents that adapt and evolve across domains.

Build the agent with enhanced reasoning and acting capabilities. The effectiveness of interactive agents rests on two tightly coupled capabilities: generating high-quality reasoning rationales and

invoking external tools when necessary. However, current training pipelines face three structural bottlenecks: scarcity of long-form reasoning supervision, fragmented modeling of active tool use, and disjoint optimization of reasoning vs. tool learning. I therefore developed a three-step capability stack that progressively closes this gap. First, I introduced Cue-CoT [2], a self-synthesis paradigm in which the model first infers latent linguistic cues — emotion, personality, psychology — from context and then conditions its response on them; for instance, given "what should I know before having a baby?" the model first derives the user's anxiety and need for specificity and only then generates a calibrated answer instead of a generic list, referred by Google DeepMind (EACL'24) as a representative mitigation for stereotyped dialogue, and extended in SRLM [3] during my internship at ByteDance-Seed. Second, given stronger reasoning, the agent also needs better acting capabilities; to that end I decoupled tool learning into a unified plan → execute → generate framework that explicitly models inter-tool dependencies (SAFARI [4], Best Paper Award @ IDF 2023). Third, I introduced Search-R2 [5] in collaboration with Tencent-Hunyuan, which advanced RL-based training of search agents with denser process supervision, leading to better search behavior and outcomes. In addition, to jointly optimize reasoning and tool use, I presented a comprehensive reward recipe for the tool-integrated reasoning training paradigm via reinforcement learning [6] — adopted and benchmarked against by IBM, NVIDIA, and Allen AI, which has received 200+ citations and 400+ GitHub stars within a year of release. Together, these three innovations establish the capability substrate — strong E_{int} and strong E_{ext}, jointly optimized — that behavioral regulation will then govern.

Regulate the agent with controllable thinking and acting behavior. Capability alone is insufficient. In complex interactive environments, reasoning and tool use are not isolated modules but alternative strategies within a single decision process. Existing systems rely on heuristic pipelines, producing blurred decision boundaries, tool overuse, and unstable interaction patterns — precisely the failure modes ToA predicts. I therefore developed a coherent control stack along three coupled axes — think-vs-act routing [7, 8], reasoning-budget allocation [9], and tool-behavior optimization [10]. In detail, I developed Self-DC [7], which is a dynamic RAG framework grounded in metacognitive self-awareness: agents use verbalized and probability-based confidence signals to decide, at each step, whether to think internally or retrieve externally. SMART [8] is the SFT counterpart of Self-DC — a metacognition-inspired training paradigm that reduces tool calls by 24% while improving performance by 37% on Llama-3.1-70B. Once the when is resolved, the next question is how much — how many reasoning tokens to spend, which tool to call, and in what manner. To this end, I developed several efficient thinking and acting algorithms to control the budget based on task difficulty and model capability, including AdaCtrl [9] and OTC-PO [10], which introduces RL-based optimal tool calling under resource constraints, featured by The AI Timeline as a representative o3-style efficiency framework. Together, these works elevate language agents from tool-enabled systems to decision-regulated ones — answering not only "can the agent solve the task" but "does it solve the task efficiently, stably, and on principled grounds.".

Diagnose the agent under realistic environments and sustained interaction. Evaluation closes the loop between training and deployment. I view agents as interactive decision-makers in dynamic environments - a view that demands evaluation along multiple, often-overlooked axes. Standard benchmarks mostly measure task success; my benchmarks measure how success is achieved — and where it breaks. I developed a multi-dimensional evaluation framework spanning three layers. For interaction dynamics, AppBench [11] benchmarks graph-structured multi-API planning, on which even GPT-4o achieves only 2% success at the hardest tier; DialogTool [12] extends this to multi-turn state evolution. For behavioral attributes, PerLTQA [13] probes long-term personal-memory QA over multi-session reasoning, which won Best Paper Award at SIGHAN workshop at ACL 2024. I further developed a proactive, pre-execution tool-safety benchmark SafeToolBench [14] with the accompanying SafeInstructTool defense framework (+17% risk resistance); the related multimodal harmful-content detector OSPC won the global championship at AI Singapore's Online Safety Prize Challenge

(WWW 2024), ranking 1st among 135 teams from 34 countries. Finally, at the cross-environment generalization layer, I built a metadata-aligned data annotation platform that automatically parses heterogeneous interaction interfaces across operating systems, applications, and versions, on top of which TransBench [15] is the first benchmark for cross-version, cross-platform, and cross-app GUI-agent transfer, and a GUI agent trained on its auto-collected data beats OS-Atlas (proposed by MIT, HKU and Shanghai AI Lab) by 5 absolute points across Android, iOS, and Web. Together, these benchmarks are not products but instruments to expose failure, directly adopted by UCLA, KCL, the Alan Turing Institute, Tencent, and Baidu — and the failure modes they expose now define my future-work agenda.

Future Work: Toward Lifelong Intelligent Agents

Building on this closed-loop foundation, my long-term goal is to advance language agents from single-phase optimization to lifelong systems that learn, align, and evolve across domains under sustained interaction. Borrowing the L1–L5 automation taxonomy from autonomous driving, today's agents operate around Level 3 (long-horizon tasks with complex tool sets); the path to Level 5 (fully autonomous self-evolution) requires both broader empirical observations across domains and three thrusts grounded in ToA.

Applications: Agent + X. The fastest way to stress-test ToA is to deploy it where domain constraints are non-negotiable. I am therefore pursuing four Agent + X lines, each chosen because its constraint structure attacks a different face of the theory: (1) **Agent + Science** targets agents that hypothesize, run experiments, and accelerate discovery — domains where success requires extreme epistemic discipline; (2) **Agent + Education** targets personalized learning and tutoring at scale; (3) **Agent + Finance** targets market analysis, investment reasoning, and risk management under strict cost and latency budgets; and (4) **Agent + Law** targets legal research and document analysis where every action must leave an audit trail — turning ToA's allocation rule from a normative criterion into a legally enforceable one. The vision is not "applying agents to domains" but letting domains reshape agent research: each domain surfaces constraints (slow feedback in science, audit trails in law, real-time cost in finance, individual learning trajectories in education) that the current ToA formulation does not yet capture, and addressing them is how the theory grows.

Thrust 1: Continual learning under evolving tools, tasks, and users. APIs evolve, interfaces change, and task distributions drift - exactly the failure mode TransBench [15] was built to expose. I will develop data-to-policy updating mechanisms and structured memory abstractions that convert noisy interaction trajectories into stable knowledge representations and transferable structured skills, controlling forgetting while bounding computational cost.

Thrust 2: Long-term alignment under distribution shift. As agents adapt, user preferences drift and environmental constraints may weaken, where alignment achieved at training time silently decays at deployment time. I will pursue two coupled directions: (i) *decomposable preference representations* that separate learnable personalization [2] (interests, style) from inviolable safety floors [14] (legality, irreversible actions), with online tracking of latent user intent and explicit modeling of environment-level constraints; (ii) *robust long-horizon alignment policies* combining internal self-constraint mechanisms with external defense against prompt injection, jailbreak chains, and adversarial tool inputs, so safety behaves as a boundary, not a suggestion.

Thrust 3: Agent evolution — restructuring skills and internal models. The two thrusts above keep an agent calibrated and aligned within a fixed architecture; but in the long run, fixed model structures and predefined tool sets become the binding constraint. Beyond using tools, future agents must compose and create new tools, internalize parts of the external environment through learned world models [16], and coordinate as multi-agent collectives. In ToA terms: an agent that internalizes external tool capability progressively collapses E_{ext} into E_{int} over time with an expanding task set.

References

[1] **Hongru Wang**, Cheng Qian, Manling Li, Jiahao Qiu, Boyang Xue, Mengdi Wang, Heng Ji, Amos Storkey, and Kam-Fai Wong. Position: Agent should invoke external tools only when epistemically necessary. In *ICML 2026 Position Paper Track* (*to appear*), 2026.

[2] **Hongru Wang**, Rui Wang, Fei Mi, Yang Deng, Zezhong Wang, Bin Liang, Ruifeng Xu, and Kam-Fai Wong. Cue-CoT: Chain-of-thought prompting for responding to in-depth dialogue questions with LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12047–12064, Singapore, December 2023. Association for Computational Linguistics.

[3] **Hongru Wang**, Deng Cai, Wanjun Zhong, Shijue Huang, Jeff Z. Pan, Zeming Liu, and Kam-Fai Wong. Self-reasoning language models: Unfold hidden reasoning chains with few reasoning catalyst. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5578–5596, Vienna, Austria, July 2025. Association for Computational Linguistics.

[4] **Hongru Wang**, Minda Hu, Yang Deng, Rui Wang, Fei Mi, Weichao Wang, Yasheng Wang, Wai-Chung Kwan, Irwin King, and Kam-Fai Wong. Large language models as source planner for personalized knowledge-grounded dialogues. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9556–9569, Singapore, December 2023. Association for Computational Linguistics.

[5] Bowei He, Minda Hu, Zenan Xu, **Hongru Wang**†, Licheng Zong, Yankai Chen, Chen Ma, Xue Liu, Pluto Zhou, and Irwin King. Search-r2: Enhancing search-integrated reasoning via actor-refiner collaboration. In *ICML 2026* (*to appear*), 2026.

[6] Cheng Qian, Emre Can Acikgoz, Qi He, **Hongru Wang**, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. ToolRL: Reward is all tool learning needs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[7] **Hongru Wang**, Boyang Xue, Baohang Zhou, Tianhua Zhang, Cunxiang Wang, Huimin Wang, Guanhua Chen, and Kam-Fai Wong. Self-DC: When to reason and when to act? self divide-and-conquer for compositional unknown questions. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies* (*Volume 1: Long Papers*), pages 6510–6525, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[8] Cheng Qian, Emre Can Acikgoz, **Hongru Wang**, Xiusi Chen, Avirup Sil, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. SMART: Self-aware agent for tool overuse mitigation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4604–4621, Vienna, Austria, July 2025. Association for Computational Linguistics.

[9] Shijue Huang*, **Hongru Wang***, Wanjun Zhong*, Zhaochen Su, Jiazhan Feng, Bowen Cao, and Yi R. Fung. Adactrl: Towards adaptive and controllable reasoning via difficulty-aware budgeting. *Transactions on Machine Learning Research*, 2026.

[10] **Hongru Wang**, Cheng Qian, Wanjun Zhong, Xiusi Chen, Jiahao Qiu, Shijue Huang, Bowen Jin, Mengdi Wang, Kam-Fai Wong, and Heng Ji. Acting less is reasoning more! teaching language model to act efficiently. In *LAW Workshop at NeurIPS 2025*, 2025.

[11] **Hongru Wang**, Rui Wang, Boyang Xue, Heming Xia, Jingtao Cao, Zeming Liu, Jeff Z. Pan, and Kam-Fai Wong. AppBench: Planning of multiple APIs from various APPs for complex user instruction. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15322–15336, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[12] **Hongru Wang**, Wenyu Huang, Yufei Wang, Yuanhao Xi, Jianqiao Lu, Huan Zhang, Nan Hu, Zeming Liu, Jeff Z. Pan, and Kam-Fai Wong. Rethinking stateful tool use in multi-turn dialogues: Benchmarks and challenges. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5433–5453, Vienna, Austria, July 2025. Association for Computational Linguistics.

[13] Yiming Du, **Hongru Wang**, Zhengyi Zhao, Bin Liang, Baojun Wang, Wanjun Zhong, Zezhong Wang, and Kam-Fai Wong. PerLTQA: A personal long-term memory dataset for memory classification, retrieval, and fusion in question answering. In Kam-Fai Wong, Min Zhang, Ruifeng Xu, Jing Li, Zhongyu Wei, Lin Gui, Bin Liang, and Runcong Zhao, editors, *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pages 152–164, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[14] Hongfei Xia*, **Hongru Wang***, Zeming Liu, Qian Yu, Yuhang Guo, and Haifeng Wang. SafeToolBench: Pioneering a prospective benchmark to evaluating tool utilization safety in LLMs. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 17643–17660, Suzhou, China, November 2025. Association for Computational Linguistics.

[15] Yuheng Lu*, Qian Yu*, **Hongru Wang***, Zeming Liu, Wei Su, Yanping Liu, Yuhang Guo, Maocheng Liang, Yunhong Wang, and Haifeng Wang. TransBench: Breaking barriers for transferable graphical user interface agents in dynamic digital environments. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12464–12478, Vienna, Austria, July 2025. Association for Computational Linguistics.

[16] Yixia Li, **Hongru Wang**†, Jiahao Qiu, Zhenfei Yin, Dongdong Zhang, Cheng Qian, Zeping Li, Pony Ma, Guanhua Chen, and Heng Ji. From word to world: Can large language models be implicit text-based world models? In *ACL 2026 (Oral, to appear)*, 2026.