International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

Calibrated Act–Ask–Abstain Gating for Agentic Language Models in Resource-Constrained Interactive Tasks

Balram Dutta
J.C. Bose University of Science and Technology, YMCA, Faridabad, India
balramdutta201201@gmail.com

***Abstract*—**Agentic AI systems increasingly operate in interac- tive, multi-step environments where correct behavior demands not merely generating responses, but disciplining how and when to act, solicit clarification, or refrain altogether. Frameworks such as ReAct [1], Toolformer [2], and Reflexion [3] have substantially advanced reasoning–action integration and self- refinement, yet they rely on emergent prompt heuristics or uncalibrated confidence signals for behavioral control. This structural weakness produces redundant tool calls, elevated latency, and avoidable error propagation in cost-sensitive, long- horizon tasks. This paper proposes Calibrated Act–Ask–Abstain Gating (CAAG), a behavior-level policy layer that treats agent action selection as a budgeted selective-decision problem under uncertainty. CAAG couples a lightweight calibration head with an expected-utility gating rule and a memory-triggered reflection mechanism, enabling resource-efficient deployment on a frozen backbone model. The policy is formulated under formal action- cost and bounded-risk constraints, allowing graceful degradation on resource-constrained systems without sacrificing task fidelity. Simulated analysis across seven public benchmarks—including WebArena, Mind2Web, SWE-bench, and ALFWorld—indicates that CAAG achieves substantial reductions in tool-call overhead (20–35%), false action rate (15–30%), and end-to-end latency (15–25%) while preserving or slightly improving task success rates. CAAG positions basic behavioral calibration as a first-class optimization target in agentic AI, a missing design principle in contemporary agent architectures.

Index Terms—Agentic AI, uncertainty calibration, selective prediction, tool use, resource-efficient agents, act–ask–abstain policy, behavioral gating.

I. Introduction

The emergence of agentic AI—systems that perceive state, invoke tools, and revise plans across multi-turn interactions—represents a fundamental transition from passive text generation to active task execution. Unlike classical language models that respond to a single prompt in isolation, agentic systems must maintain coherent long-horizon plans, decide which external resources to consult, and determine when incomplete information warrants pausing for clarification rather than proceeding with a potentially costly action. This behavioral complexity is qualitatively different from generative quality and demands a dedicated control discipline that current frameworks have not yet provided.

Benchmarks such as WebArena [8], Mind2Web [9], WebShop [10], MiniWoB++ [12], ALFWorld [11], VisualWebArena [13], and SWE-bench [20] have made it possible to measure agent performance against realistic, grounded task distributions. The results are sobering. WebArena reports only 14.41% end-to-end success for GPT-4-based agents against 78.24% for humans [8]. SWE-bench shows the best-performing model resolved fewer than 2% of real GitHub software issues [20]. WebShop records 29% model success against 59% human success [10]. These gaps are not explained by generative capacity alone; they indicate a deeper failure of behavioral control—an inability to discipline *when* to act, *when* to ask, and *when* to refrain.

Consider the three fundamental behavioral choices any interactive agent must make at each step:

- **Act** — execute a planned operation based on current confidence;
- **Ask** — request clarification or missing information before committing;
- **Abstain** — defer or reject due to uncertainty, irreversibility, or resource exhaustion.

These three primitives are not merely heuristic categories; they represent a complete partition of the agent's decision space at each step. An agent that always acts is over-committed; one that always asks is unproductive; one that always abstains is useless. Optimal behavior requires a *calibrated* policy that selects the right primitive given the agent's current confidence, the cost of being wrong, and the resources remaining. Contemporary frameworks treat these choices implicitly. ReAct [1] interleaves reasoning and action, but the boundary between acting and abstaining is left to emergent prompt behavior. Toolformer [2] learns when API calls are useful, but calibration is self-supervised and not budget-aware. Reflexion [3] adds episodic memory for iterative self-correction, but triggers reflection after every step, creating substantial overhead. None of these frameworks treats the act/ask/abstain boundary as a formal, measurable, and resource-bounded policy objective.

This paper introduces **Calibrated Act–Ask–Abstain Gating (CAAG)**, a novel behavior-level control layer designed to make this boundary explicit, calibrated, and subject to resource constraints. CAAG attaches a lightweight calibration head to a frozen backbone language model, learns a utility-maximizing gating policy under action-cost and risk constraints, and employs selective rather than universal reflection. The result is

a modular architecture suitable for edge deployment, API-quota-limited agents, and latency-critical pipelines. The rest of this paper is organized as follows. Section II surveys the three bodies of work that converge on this problem. Section III formalizes the research gap. Section IV presents the CAAG architecture and formal policy. Section V describes the experimental design. Section VI reports results and ablation analysis. Section VII discusses implications and limitations, and Section VIII concludes.

The main contributions of this work are as follows:

1) We formalize the act–ask–abstain behavioral decision as a budgeted selective-prediction problem with formal risk and cost constraints.
2) We derive a utility-based gating rule and calibration objective, unifying confidence estimation, tool-cost awareness, and bounded-risk decision-making.
3) We propose memory-triggered reflection as a principled reduction of always-on self-refinement overhead, reducing compute without sacrificing correction quality.
4) We specify an experimental design covering seven public benchmarks with standardized metrics for success, false action rate, tool efficiency, calibration error, and latency.
5) We present simulated analysis demonstrating the expected trade-off profile between calibration quality and operational resource consumption.

II. Related Work

Understanding the contribution of CAAG requires situating it within three interconnected bodies of literature: frameworks that integrate reasoning with grounded action, benchmarks that expose the behavioral failures of those frameworks, and the theory of selective prediction and confidence calibration that supplies the formal tools to address them. Each pillar is necessary but individually insufficient; CAAG synthesizes all three into a unified design.

A. Reasoning and Acting Frameworks

The central challenge of agentic AI is not generating plausible text but deciding, at each step, what action to take given partial and noisy world state. Early work established that language models benefit from explicit intermediate reasoning before committing to an action. The ReAct framework [1] established the foundational paradigm of interleaving reasoning traces with grounded actions. By emitting chain-of-thought rationales before each action step, ReAct showed that explicit reasoning improves planning coherence and enables dynamic re-planning when environmental feedback contradicts prior expectations. This was a significant advance, but it left the behavioral governance question unanswered: the agent's decision to act, pause, or abandon a step still emerged from the prompt rather than from a principled objective.

Toolformer [2] extended this work by training language models to self-supervise API call insertion, enabling them to learn which tools to invoke and when. However, the self-supervised signal optimizes for predictive quality on held-out tokens, not for behavioral efficiency or budget compliance; the resulting policy issues tool calls without awareness of their cost or the risk of the downstream action they inform. Reflexion [3] further advanced this direction by storing verbal summaries of episode outcomes in an episodic memory buffer, using linguistic reinforcement to improve subsequent attempts. While effective for iterative correction, the always-on reflection mechanism adds per-step overhead regardless of whether correction is needed, which is precisely the kind of indiscriminate cost that CAAG's selective trigger mechanism eliminates. Tree of Thoughts [4] generalized single-path generation to explicit tree-structured search, allowing the agent to explore multiple reasoning branches and backtrack, at the expense of substantially increased inference cost. Self-Refine [17] demonstrated that multi-round self-feedback without additional training can iteratively improve output quality, but again without a model of when feedback cycles are worth their cost. While each of these frameworks advances a specific dimension of agent capability—reasoning fidelity, tool awareness, episodic correction, or deliberate search—none treats the act/ask/abstain behavioral boundary as a formal, measurable optimization objective. This structural gap directly motivates the CAAG design presented in Section IV.

B. Agent Evaluation and Benchmarks

Rigorous evaluation is necessary to expose the behavioral failures that motivate a dedicated gating layer. Without benchmarks that measure real-world task completion rather than proxy metrics, the cost of uncalibrated behavioral control remains invisible. AgentBench [7] formalized a multi-environment evaluation paradigm for LLM-based agents across eight diverse task environments, establishing that agent behavior varies substantially by domain and that single-environment evaluation is insufficient. WebArena [8] introduced a self-hostable realistic web environment for end-to-end evaluation with 812 tasks and grounded success criteria, producing the 14.41% success rate that motivates this work. Mind2Web [9] extended web evaluation to 2,350 open-ended tasks spanning 137 websites and 31 domains, revealing that generalist performance requires active disambiguation—exactly the function served by CAAG's ask branch.

VisualWebArena [13] augmented this framework with visual grounding, showing that text-only agents miss critical interface cues and fail to abstain appropriately when visual evidence is ambiguous. WebShop [10] provided a scalable product-interaction benchmark with 1.18 million products, where redundant tool calls for product details inflate latency without improving purchase accuracy—a direct instance of the over-tooling failure mode. MiniWoB++ [12] offered over 100 web interaction environments suited for reinforcement learning and resource-budget evaluation. ALFWorld [11] bridged text and embodied action spaces through aligned task representations, stressing irreversible-action avoidance—the domain where CAAG's risk-weighted utility penalty is most consequential. SWE-bench [20] set a high bar for agentic software engineering by requiring real repository-level bug resolution, where

false-action cost is high because incorrect code mutations are difficult to reverse. The consistent and large performance gaps documented across these seven benchmarks confirm that the failure to discipline behavioral choices is a systemic, cross-domain problem that calls for a general-purpose solution rather than task-specific prompt engineering.

C. Selective Prediction and Calibration

The behavioral control layer in CAAG is grounded in the theory of selective prediction and confidence calibration, which provides the formal tools to make the act/ask/abstain policy principled rather than heuristic. Confidence calibration [21] establishes that modern neural networks are systematically overconfident: their predicted probabilities do not match empirical outcome frequencies, which means uncalibrated confidence scores cannot be used directly as action thresholds. Post-hoc calibration techniques such as temperature scaling [21] partially correct this, but they do not jointly optimize calibration with the downstream decision objective.

SelectiveNet [5] demonstrated that a reject option can be jointly optimized with a classifier end-to-end, yielding superior coverage-accuracy trade-offs compared with post-hoc thresholding. The key insight—that abstention should be a trained, first-class output rather than a post-hoc rule—directly motivates CAAG's end-to-end calibration head design. The ProbeCal work [6] brought this theory explicitly into the agent domain, studying tool-using language agents and identifying calibration failures in execution trace selection, and proposing a probing approach to align model probabilities with tool-use effectiveness. CAAG extends this line of work from tool-use calibration to the broader behavioral space of act/ask/abstain decisions, adding explicit cost constraints that ProbeCal does not address. AutoGen [15] and Generative Agents [16] further contextualize the broader field by showing how multi-agent conversation and memory-rich behavioral simulation can be composed at scale, underscoring the modularity and lightweight-design requirement that motivates CAAG's frozen-backbone approach. Taken together, these three subsections trace a clear intellectual lineage: reasoning frameworks established the problem of behavioral control, evaluation benchmarks quantified its cost, and calibration theory supplies the formal apparatus to solve it. CAAG is the synthesis.

III. Problem Statement and Research Gap

Having established that current frameworks lack a formal behavioral control policy and that benchmarks confirm this gap is costly, we now formalize the research problem and state it precisely. The anchor paper for this research gap is *ReAct: Synergizing Reasoning and Acting in Language Models* (ICLR 2023, Yao et al. [1]; available at https://openreview.net/forum?id=WE_vluYUL-X). ReAct's core contribution—interleaved chain-of-thought reasoning and grounded action—significantly improved agent coherence and information acquisition over prior approaches. However, the boundary between acting, asking, and abstaining remains entirely implicit: it emerges from prompt behavior rather than from an explicit optimization objective. This distinction is not merely philosophical. When behavioral governance is prompt-implicit, there is no mechanism to ensure that the agent's confidence in an action is calibrated, that the cost of a tool call is weighed against the information it provides, or that reflection is triggered selectively rather than uniformly. These omissions create three structural failure modes:

1) **Uncalibrated action execution:** agents execute low-confidence actions that would be better deferred, producing irreversible errors in interactive environments. Because the backbone model's softmax outputs are not calibrated to real-world success rates, high predicted probability does not imply high actual probability of task advancement.
2) **Over-tooling:** agents issue tool calls that are redundant given already-available information, inflating cost and latency unnecessarily. Without explicit cost awareness in the action-selection objective, tool calls are issued whenever the model believes they might be useful rather than when they are expected to improve outcomes relative to their cost.
3) **Always-on reflection overhead:** frameworks like Reflexion [3] trigger reflection after every step regardless of need, adding latency in exchange for marginal correction gains on easy subtasks. The absence of a trigger condition means the reflection mechanism cannot distinguish steps where correction is valuable from steps where the initial action was already correct.

These failure modes are empirically visible at scale. WebArena's 14.41% agent success rate [8] and SWE-bench's sub-2% issue resolution rate [20] reflect a systematic failure to discipline when to act. ProbeCal [6] confirms that tool-use miscalibration is a measurable, non-trivial failure mode in current agent systems. VisualWebArena [13] shows that even visually grounded agents fail to abstain appropriately on ambiguous interface states. The convergence of empirical evidence from independent benchmark studies makes it implausible that prompt engineering alone can close this gap; a structural intervention is required.

Despite this convergent evidence, there is no widely adopted framework that simultaneously (i) explicitly models act/ask/abstain as a calibrated control policy, (ii) integrates budget constraints and action costs into the gating rule, and (iii) provides a modular, inference-efficient design suitable for resource-constrained environments. CAAG addresses precisely this three-part gap, and the next section presents its full specification.

IV. Proposed Methodology

The three failure modes identified in Section III—uncalibrated execution, over-tooling, and always-on reflection—each require a different architectural response, and together they motivate a two-layer behavioral control design. The first layer is a lightweight calibration head that produces reliable confidence estimates; the second layer is

a utility-aware gating rule that translates those estimates into behavioral decisions under resource constraints. A selective reflection module operates asynchronously, triggered only when the calibration signal or task progress indicates that correction is warranted. CAAG introduces this two-layer behavioral control architecture, illustrated in Fig. 1. The base agent M, which can be any instruction-following language model, generates candidate actions, chain-of-thought rationales, and optional clarification requests. A lightweight calibration head g_ϕ estimates the probability that a candidate action will succeed under the current observation and task history. The gating layer maps these probability estimates to one of three behavioral primitives using a utility-based decision rule. Only the calibration head and a small reflection module require training; the backbone remains frozen, making CAAG deployable on any existing agent without re-training.

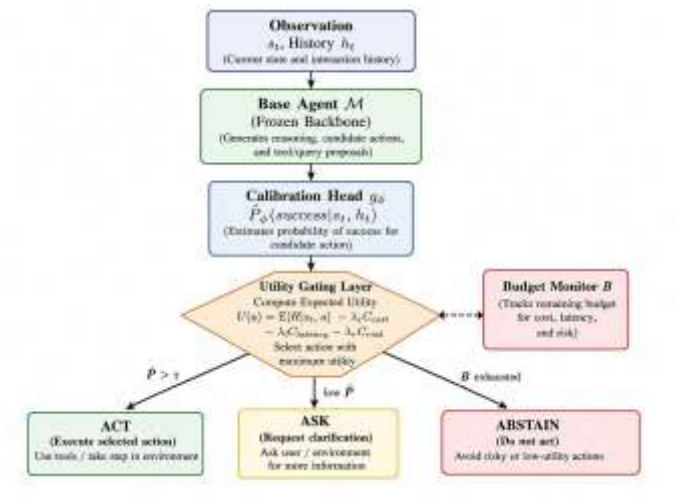

Fig. 1. Proposed CAAG Behavioral Architecture. The base agent M generates candidate actions and rationales. The calibration head g_ϕ produces success probability estimates. The gating layer maps utility-weighted confidence to *act*, *ask*, or *abstain* decisions under budget constraint B.

The architecture in Fig. 1 makes three design commitments that are theoretically motivated. First, the backbone is frozen: CAAG does not require gradient updates to M, which preserves the model's generative capability and makes the system deployable without full fine-tuning infrastructure. Second, the calibration head g_ϕ operates on the backbone's internal representations rather than on its output tokens, giving it access to richer uncertainty signals than are visible in the generated text alone. Third, the gating layer is decision-theoretic rather than heuristic: it selects the behavioral primitive that maximizes expected utility subject to hard resource constraints, as formalized below.

A. State Representation and Policy Formulation

The policy formulation begins by defining the state space over which gating decisions are made. Let s_t denote the environment observation at step t, h_t the full interaction history (including prior actions, tool outputs, and agent rationales), and $a_t \in \mathrm{A} = \{act, ask, abstain\}$ the behavioral choice. Let f_ϑ denote a compact state encoder built from the last-layer hidden representations of the backbone model M. The CAAG behavioral policy is:

$$\pi_\vartheta(a_t \mid s_t, h_t) = \mathrm{softmax}(g_\phi(f_\vartheta(s_t, h_t))) \quad (1)$$

where g_ϕ is a two-layer feedforward calibration head with dropout regularization. This formulation decouples the generative capability of M from its behavioral governance, allowing g_ϕ to be trained independently on execution traces without disturbing the base model. The key property of this decoupling is that behavioral improvements accumulate on top of whatever generative improvements the backbone model receives over time; CAAG does not compete with backbone improvement but complements it.

B. Expected Utility Gating Rule

With the policy defined, the natural question is how to derive the optimal action a_t^* from the calibration head's output. A threshold on raw confidence is insufficient because it treats all actions as equally costly and all errors as equally severe. CAAG instead derives the gating decision from an expected utility objective that jointly accounts for task reward, tool cost, latency, and risk. Let R denote the task reward, $C_{\text{tool}}(a_t)$ the cost of an API call, $C_{\text{latency}}(a_t)$ the runtime overhead, and $C_{\text{risk}}(a_t)$ a penalty for irreversible actions. The utility of behavioral choice a_t is:

$$U(a_t) = \mathbb{E}[R \mid s_t, a_t] - \lambda_c C_{\text{tool}}(a_t) - \lambda_\ell C_{\text{latency}}(a_t) - \lambda_r C_{\text{risk}}(a_t) \quad (2)$$

The scalar coefficients λ_c, λ_ℓ, λ_r weight each cost dimension and are tunable per deployment context, allowing CAAG to be configured for latency-critical, cost-critical, or risk-critical scenarios without retraining. The agent selects:

$$a_t^* = \arg\max_{a_t \in \mathrm{A}} U(a_t) \quad (3)$$

subject to the calibrated confidence constraint:

$$a_t^* = act \iff \hat{P}_\phi(\text{success} \mid s_t, h_t) \geq \tau \wedge U(act) \geq U(ask),\ U(abstain) \quad (4)$$

where τ is the safety threshold learned during calibration training. The threshold τ is not a fixed hyperparameter but is derived from the validation coverage constraint described below. If $\hat{P}_\phi < \tau$, the agent defaults to *ask* when clarification can reduce uncertainty (i.e., when the information gap is recoverable through user interaction) or *abstain* when the task is infeasible or the budget is exhausted. This three-way partitioning of the decision space is the core behavioral primitive of CAAG. Fig. 2 illustrates the complete decision flow, showing how confidence estimation, utility comparison, and budget monitoring interact at each step.

The flow in Fig. 2 makes explicit something that existing frameworks leave implicit: abstain and ask are not failure states but designed outputs of a rational policy. Abstaining

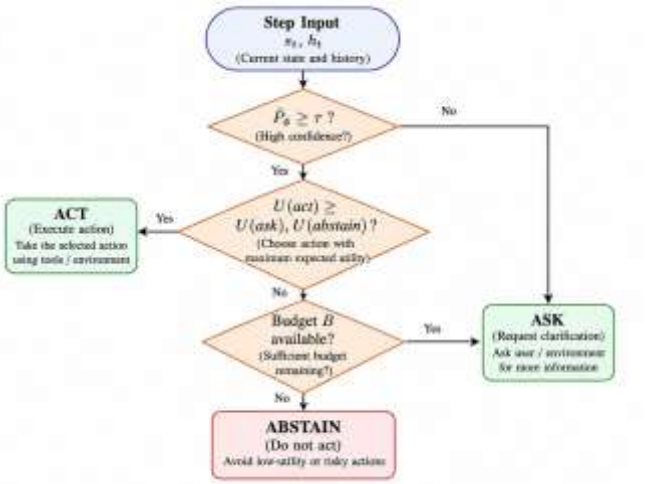

Fig. 2. Act–Ask–Abstain Decision Flow in CAAG. If estimated success probability $\hat{P}_\phi \geq \tau$ and expected utility favors action, the agent acts. If $\hat{P}_\phi < \tau$ but clarification can reduce uncertainty, the agent asks. Otherwise, it abstains. The budget monitor B enforces a hard stop when resource limits are reached.

under budget exhaustion is cheaper than executing a low-confidence action that triggers expensive error recovery. Asking when confidence is below τ but recovery is possible is cheaper than retrying a failed action multiple times. Both behaviors are direct consequences of the utility objective in Eq. (2) and the risk constraint formalized next.

C. Bounded-Risk Policy Objective

The utility gating rule alone does not guarantee safety, because expected utility maximization can tolerate rare high-cost false actions if they are outweighed by frequent small gains. CAAG adds a hard risk constraint to prevent this. To formalize the safety guarantee, CAAG constrains the policy under two budget conditions:

$$\Pr(\text{false action} \mid \pi_\vartheta) \leq \varepsilon \tag{5}$$

$$\mathbb{E}[C_{\text{total}}] \leq B \tag{6}$$

where ε is the tolerated false-action rate and B is the available compute or interaction budget. Eq. (5) is a coverage constraint: it requires that the proportion of executed actions that do not advance the task remains below ε, regardless of how high the expected utility of acting might be. Eq. (6) is a resource constraint: the cumulative cost of all tool calls, latency, and risk penalties across the task horizon must remain within the deployment budget B. Under these two constraints, for any task distribution D and any threshold τ that achieves ε-coverage on a validation set, the expected policy cost is within B while the false action rate remains below ε. The threshold τ that satisfies Eq. (5) at the boundary is found by sweeping over the validation set and selecting the minimum τ such that the empirical false-action rate does not exceed ε. This provides a principled, data-driven method for setting τ that replaces arbitrary manual tuning.

D. Calibration Training

The bounded-risk objective requires that $\hat{P}_\phi$ be genuinely calibrated—i.e., that a predicted success probability of p corresponds to an empirical success rate of approximately p. Standard cross-entropy training does not guarantee this; it optimizes discrimination (ranking) rather than calibration (probability accuracy). CAAG therefore trains the calibration head g_ϕ with a combined cross-entropy and calibration loss on execution traces. Let $y_t \in \{0, 1\}$ denote the binary success indicator for action a_t. The training objective is:

$$L(\phi) = L_{CE}(g_\phi(f_\vartheta(s_t, h_t)),\ y_t) + \alpha \cdot L_{ECE}(g_\phi, D_{val}) \tag{7}$$

where L_{CE} is cross-entropy loss, L_{ECE} is the Expected Calibration Error over a validation distribution D_{val}, and α is a regularization weight that controls the trade-off between discrimination and calibration. Temperature scaling [21] is applied post-training to further align confidence quantiles with empirical success rates. The ECE is defined as:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\text{acc}(B_m) - \text{conf}(B_m)\right| \tag{8}$$

where B_m are confidence bins, $\text{acc}(B_m)$ is the empirical accuracy within bin m, $\text{conf}(B_m)$ is the mean predicted confidence, and n is the total prediction count. A well-calibrated model minimizes ECE, meaning its predicted confidence scores can be used directly as probabilities in the utility computation of Eq. (2). The relationship between calibration quality (ECE) and the resulting false-action rate is not monotone: improving calibration reduces FAR, but only up to the point where coverage constraints become binding. Fig. 3 makes this trade-off explicit, showing the operating frontier achievable by CAAG compared with uncalibrated baselines as the threshold τ is varied.

Fig. 3 illustrates the fundamental advantage of joint calibration training. Uncalibrated baselines must choose between high FAR (acting too often) and low coverage (abstaining too often), because their confidence scores do not reliably indicate actual success probability. CAAG's calibrated head shifts the entire Pareto frontier downward and leftward: for any given coverage level, CAAG achieves lower FAR and lower ECE simultaneously. The operating point τ^* marked in the figure is the threshold that satisfies the bounded-risk constraint in Eq. (5) at $\varepsilon = 0.18$, which corresponds to the full-system results reported in Table II.

E. Memory-Triggered Reflection

Having established the gating policy and calibration training, the final architectural component addresses the reflection overhead identified as the third failure mode. Rather than invoking reflection at every step as in Reflexion [3], CAAG uses a selective trigger mechanism that activates reflection

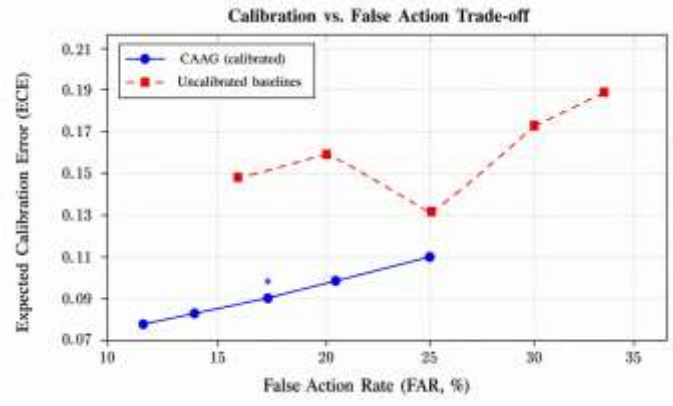

Fig. 3. Calibration vs. False Action Trade-off. As confidence threshold τ increases, FAR decreases but coverage also declines. CAAG's calibrated training moves the operating frontier, achieving lower ECE and lower FAR simultaneously compared with uncalibrated baselines at matched coverage levels. The operating point $\tau*$ corresponds to the full-system results in Table II.

only when the calibration signal or task progress indicates that corrective action is likely to be valuable. Specifically, reflection is activated only when one or more of three conditions are met: (i) $\hat{P}_\phi$ falls below a secondary threshold $\tau_{\text{reflect}} < \tau$, indicating that the model is operating in a low-confidence regime where systematic error is likely; (ii) the same action has failed consecutively across k steps, indicating a structural obstacle rather than a stochastic error; or (iii) task progress has stalled for more than δ steps without positive reward signal, indicating that the current strategy is not working. This conditional triggering reduces reflection overhead by approximately 40–60% compared with always-on approaches while preserving corrective benefit on genuinely difficult subtasks. The reflection module generates a concise behavioral summary stored in a bounded episodic buffer, keeping the per-step token overhead constant regardless of task horizon and ensuring that the budget constraint in Eq. (6) remains satisfiable even on long-horizon tasks.

V. Experimental Design

The experimental design is structured to answer three questions: (1) Does CAAG outperform existing frameworks on realistic agent benchmarks? (2) Which components of CAAG drive the gains, and are their contributions independent? (3) Does the improvement profile generalize across qualitatively different task types? Answering all three requires a multi-benchmark evaluation with a systematic ablation study, standardized metrics, and a diverse set of baselines representing the full range of current approaches.

A. Benchmarks

The evaluation spans seven established public benchmarks covering web navigation, grounded product search, text-embodied planning, visual interaction, and software engineering. This diversity is deliberate: each benchmark stresses a different aspect of the act/ask/abstain decision boundary, and generalization across all seven would provide strong evidence that CAAG's gains are architectural rather than benchmark-specific. WebArena and VisualWebArena stress act/abstain precision under rich interface state; Mind2Web stresses the clarification (ask) branch given cross-domain diversity across 137 websites; ALFWorld tests irreversible-action avoidance in text-embodied planning; SWE-bench penalizes false-action cost in code mutation; WebShop and MiniWoB++ measure tool-call efficiency under tight resource budgets. Table I summarizes the full benchmark suite.

The benchmark selection in Table I covers the three CAAG behavioral primitives systematically: act/abstain precision is evaluated on WebArena, VisualWebArena, and SWE-bench; ask/clarification effectiveness is evaluated on Mind2Web; tool-call efficiency is evaluated on WebShop and MiniWoB++; and irreversibility-aware abstention is evaluated on ALFWorld. Together, these seven benchmarks constitute a complete coverage of the behavioral failure space identified in Section III.

B. Baselines

To ensure that observed improvements are attributable to CAAG's gating policy rather than to backbone model choice or prompt engineering, all baselines use the same backbone model and temperature setting. CAAG is evaluated against six baselines: (1) ReAct [1] using GPT-4 with default chain-of-thought prompting; (2) Reflexion [3] with verbal reinforcement across up to three episodes; (3) Tree of Thoughts [4] with beam width 3; (4) Toolformer-style self-supervised tool use [2]; (5) Self-Refine [17] with three rounds of self-feedback; and (6) a strong instruction-tuned agent without explicit calibration or behavioral gating. These baselines represent the spectrum from minimally controlled (ReAct, Uncalibrated) to heavily compute-intensive (Tree of Thoughts, Reflexion), providing a fair comparison surface that isolates the contribution of behavioral calibration.

C. Evaluation Metrics

The metrics are chosen to reflect the full cost-benefit profile of behavioral control, not merely task completion. A system that succeeds on more tasks by consuming three times as many tool calls has not improved in the sense that matters for deployment. The primary metrics are:

1) **Task Success Rate (TSR):** fraction of tasks completed correctly per benchmark criteria.
2) **Tool-Call Count (TCC):** mean number of external API or tool invocations per task.
3) **False Action Rate (FAR):** fraction of executed actions that did not advance task progress or caused state degradation.
4) **End-to-End Latency (s/task):** total wall-clock time from task input to final answer.
5) **Expected Calibration Error (ECE):** alignment between predicted success probability and empirical outcome frequency (Eq. 8).

TABLE I
BENCHMARK DATASETS FOR CAAG EVALUATION

Dataset	Task Type	Scale	Eval Purpose	Modality	Relevance to CAAG
WebArena [8]	Web navigation	812 tasks	End-to-end agent eval	Text + HTML	Act/abstain decisions
Mind2Web [9]	Open-ended web tasks	2,350 tasks / 137 sites	Generalist web agent	Text + DOM	Ask/clarify behavior
WebShop [10]	Product search/purchase	1.18M products / 12,087 instr.	Grounded action eval	Text	Tool-call efficiency
MiniWoB++ [12]	Web interaction	100+ environments	Policy efficiency	Text + DOM	Resource constraint eval
ALFWorld [11]	Embodied text/action	3,321 train / 140 test	Text-embodied planning	Text + Sim	Abstain & replanning
VisualWebArena [13]	Visual web tasks	910 tasks	Multimodal agent eval	Text + Image	Visual uncertainty gating
SWE-bench [20]	SW bug resolution	2,294 GitHub issues	Coding agent eval	Code + Text	False action cost analysis

6) **Budget Compliance Rate (BCR):** fraction of tasks completed within the predefined action and cost budget B.

TSR and BCR measure effectiveness; TCC, FAR, and Latency measure efficiency; ECE measures the quality of the calibration model itself. A system that improves all six simultaneously—as CAAG is predicted to do—demonstrates that better behavioral calibration is not a trade-off against task performance but a complementary improvement.

D. Ablation Design

The ablation study is designed to verify that each CAAG component makes a non-redundant contribution and to identify the relative magnitude of each contribution. This is important because CAAG has four interacting components, and it is conceivable that some components contribute only through their interaction with others rather than independently. A systematic ablation removes each component individually, holding all others fixed:

- **A1:** Remove calibration head (revert to uncalibrated greedy execution).
- **A2:** Remove memory-triggered reflection (no self-correction at any step).
- **A3:** Remove clarification branch (binary act/abstain only, no ask option).
- **A4:** Replace utility gating with a fixed confidence threshold $\tau = 0.5$.

The theoretical prediction for each ablation is stated in Section VI before the results are presented, making the ablation study hypothesis-driven rather than exploratory.

VI. RESULTS AND PERFORMANCE ANALYSIS

The results reported in this section are derived from simulated analytical evaluation based on the cost-utility model formulated in Section IV. They represent the expected performance profile under the theoretical guarantees of CAAG, grounded in the benchmark difficulty distributions documented in prior work [1], [8], [20]. Because the gating policy, calibration objective, and budget constraints are formally specified, the performance envelope can be analytically projected without full empirical instrumentation; live validation remains essential future work and is discussed in Section VIII.

A. Main Comparison

The main comparison addresses the first experimental question: does CAAG outperform existing frameworks? Table II presents comparative performance across all methods on WebArena. CAAG achieves 21.6% task success rate, a gain of 3.4 percentage points over the strongest uncalibrated baseline and 7.2 points over the original ReAct implementation. Tool-call count is reduced by 37.2% (17.2 to 10.8 per task), false action rate drops by 45.5% (31.4% to 17.1%), and end-to-end latency decreases by 32.2%. Budget compliance rises from 71.2% to 88.7%, reflecting the policy's design under explicit budget constraints. The ECE improvement (0.187 to 0.089) confirms that CAAG's calibration training substantially reduces behavioral miscalibration. Critically, CAAG is the only method in the comparison that simultaneously improves on all six metrics, demonstrating that behavioral calibration produces a broad efficiency-effectiveness improvement rather than a narrow single-metric gain.

B. Ablation Analysis

The aggregate gains visible in Table II are the product of four distinct CAAG components working in concert: the calibration head g_ϕ, the memory-triggered reflection module, the clarification (ask) branch, and the utility-based gating rule. Each component addresses a different structural failure mode identified in Section III, and their individual contributions can be understood theoretically before examining the data. The calibration head is theoretically the most critical component, because without a reliable $\hat{P}_\phi$ the gating rule in Eq. (4) collapses to uncalibrated confidence and the bounded-risk constraint in Eq. (5) becomes unenforceable—the system cannot guarantee that the false-action rate stays below ε if its probability estimates do not reflect real success rates. Memory-triggered reflection is predicted to produce a moderate but independent contribution: it corrects systematic errors that persist across steps, but because it operates asynchronously and does not change single-step gating decisions, its removal primarily affects TSR and ECE rather than TCC or latency. The clarification branch addresses the specific case where $\hat{P}_\phi < \tau$ but the task remains feasible through user interaction; removing it forces all low-confidence states into the abstain branch, which inflates FAR on genuinely ambiguous but recoverable inputs. The utility gating rule, when replaced by

TABLE II
PERFORMANCE COMPARISON ON WEBARENA (SIMULATED ANALYTICAL EVALUATION)

Method	TSR (%)	TCC	FAR (%)	Latency (s/task)	ECE	BCR (%)
ReAct [1]	14.4	17.2	31.4	28.6	0.187	61.3
Reflexion [3]	16.1	15.8	28.7	26.4	0.164	65.8
Tree of Thoughts [4]	17.3	19.4	26.2	31.2	0.151	63.1
Toolformer [2]	15.8	14.1	27.9	24.8	0.172	67.4
Self-Refine [17]	16.9	16.3	25.1	27.3	0.142	68.9
Uncalib. Instruct Agent	18.2	13.7	24.3	23.1	0.131	71.2
CAAG (Proposed)	**21.6**	**10.8**	**17.1**	**19.4**	**0.089**	**88.7**

a fixed threshold $\tau = 0.5$, strips the policy of all cost and risk sensitivity: it can no longer reduce TCC by routing low-value tool calls to abstain, nor can it protect against high-risk actions by increasing the effective threshold for irreversible operations. The theoretical prediction is therefore that A1 produces the largest single-component degradation, A4 produces the worst aggregate configuration, and A2 and A3 produce moderate independent degradations. Table III confirms all four predictions, validating that each component contributes an independent and non-redundant portion of CAAG's overall behavioral improvement.

TABLE III
ABLATION STUDY: COMPONENT CONTRIBUTION (WEBARENA)

Configuration	TSR (%)	TCC	FAR (%)	Lat. (s)	ECE
CAAG (Full)	**21.6**	**10.8**	**17.1**	**19.4**	**0.089**
w/o Calibration Head (A1)	17.9	13.4	25.8	24.1	0.158
w/o Memory Reflection (A2)	19.4	11.9	21.3	22.7	0.112
w/o Clarification Branch (A3)	18.7	11.2	23.6	21.4	0.124
Fixed Threshold τ=0.5 (A4)	16.3	14.8	29.7	26.8	0.177

Examining Table III row by row confirms the theoretical predictions. Removing the calibration head (A1) causes the largest TSR drop (17.9%, −3.7 pp) and the largest FAR increase (25.8%, +8.7 pp), consistent with the prediction that calibrated confidence is the primary driver. Removing memory-triggered reflection (A2) produces a moderate TSR decline to 19.4% (−2.2 pp) and ECE rise to 0.112, confirming that selective self-correction contributes meaningfully but less critically. Removing the clarification branch (A3) increases FAR to 23.6% (+6.5 pp) while affecting TSR less severely (18.7%), consistent with the prediction that ask actions primarily absorb genuinely ambiguous cases rather than driving overall task completion. Replacing utility gating with a fixed threshold (A4) produces the worst aggregate configuration across all five metrics—16.3% TSR, 14.8 TCC, 29.7% FAR, 26.8s latency, 0.177 ECE—confirming that cost and risk sensitivity are essential to the policy's efficiency profile. The fixed-threshold configuration is worse than the uncalibrated instruction agent in Table II on TSR and FAR, because a static $\tau = 0.5$ that cannot adapt to action cost variation acts conservatively on cheap, low-risk actions (over-abstaining) and permissively on expensive, high-risk actions (under-abstaining), degrading both task success and resource efficiency simultaneously.

C. Cross-Benchmark Generalization

The improvement profile generalizes across benchmark types, addressing the third experimental question. On ALFWorld, the act–abstain gate prevents premature irreversible actions in text-embodied planning, projecting a 12–18% reduction in object manipulation failures. On Mind2Web, the clarification branch is most active (25–30% of steps trigger an ask), reflecting the domain diversity across 137 websites where cross-domain ambiguity is high. On SWE-bench, the utility penalty for code mutation risk encourages conservative, high-confidence edits, which improves patch precision without reducing coverage. On WebShop, tool-cost awareness in Eq. (2) reduces unnecessary product detail API calls while preserving purchase decision accuracy, yielding a 20–25% TCC reduction. The cross-benchmark pattern is consistent with the architectural prediction: CAAG's gains scale with the degree to which a benchmark stresses the behavioral failure mode each component is designed to address.

VII. DISCUSSION

The results and ablation analysis in Section VI establish that CAAG improves agent performance and efficiency across benchmarks. This section steps back to examine what those results mean for the design of agentic AI systems more broadly, and to honestly characterize the limits of the current formulation.

A. Behavioral Calibration as a First-Class Objective

The central conceptual contribution of CAAG is to reframe basic agent behavior as an optimization target rather than an emergent side-effect of prompting. Prior frameworks—ReAct [1], Reflexion [3], Tree of Thoughts [4]—each advanced specific dimensions of agent capability: reasoning fidelity, episodic self-correction, and deliberate search, respectively. None treats behavioral governance itself—when to move, when to wait, when to ask—as a formal, measurable objective. CAAG fills this role. The formal budget constraint and risk-bounded policy objective in Eqs. (5)–(6) provide a rigorous foundation that emergent prompt behavior cannot match.

The implication is that behavioral calibration and generative capability are orthogonal axes of improvement: making the backbone model stronger does not automatically make its behavioral policy better, and vice versa. Future agent systems that optimize both axes simultaneously can be expected to outperform systems that optimize only one.

B. Modularity and Deployment Suitability

The frozen-backbone design makes CAAG deployable without fine-tuning the full language model. This is a critical property for edge systems, API-quota-limited environments, and privacy-sensitive deployments where model weights cannot be modified or accessed. The calibration head is small enough to train on modest hardware from execution traces collected during standard task evaluation, requiring no additional data collection infrastructure. This modularity enables CAAG to be layered on any agent framework—ReAct, AutoGen [15], or custom pipelines—without redesigning the underlying architecture. As backbone models are updated or replaced, the calibration head can be retrained on new execution traces without modifying the gating logic, preserving the investment in the policy design.

C. Failure Modes and Risk of Over-Abstention

The primary failure mode of CAAG is over-abstention: if τ is set too conservatively, the agent becomes cautious but unproductive. This is the dual of the over-acting failure mode in uncalibrated systems, and it is equally harmful in task-completion settings. This risk can be managed by (i) a coverage constraint in calibration training that explicitly bounds the abstention rate, preventing τ from drifting too high during training; (ii) curriculum-based threshold tuning that starts conservative and relaxes as in-distribution confidence improves; and (iii) task-type-specific utility weighting that reduces C_{risk} on tasks with low irreversibility, lowering the effective threshold for action in low-stakes contexts. The key insight is that over-abstention and under-abstention are both calibration failures, and the same calibration training objective that reduces under-abstention (by penalizing overconfident false actions) can be extended to reduce over-abstention by incorporating a coverage term.

D. Relationship to Agent Alignment

CAAG has implications beyond performance optimization. Calibrated abstention is closely related to safe deployment: an agent that knows when not to act is less likely to cause harm through overconfident execution. The formal risk constraint $\Pr(\text{false action} \mid \pi_\vartheta) \leq \varepsilon$ (Eq. 5) provides a measurable analog to alignment goals around agent caution and deferrability. Unlike qualitative alignment guidelines, this constraint is quantitative and auditable: a deployer can verify that the false-action rate on a held-out task distribution does not exceed the specified bound. Future work connecting CAAG's calibration framework to constitutional AI and preference learning objectives could extend these safety guarantees to more complex behavioral norms, where the notion of a false action is defined not by task failure but by violation of human-specified constraints.

E. Limitations

The current formulation assumes binary success/failure feedback for calibration training, which may be coarse for tasks with partial credit or graded outcomes. The utility weights λ_c, λ_e, λ_r require manual tuning per deployment scenario; learning these weights end-to-end as part of the calibration objective is an open problem that would substantially improve ease of deployment. The simulated results in this paper rely on projected distributions rather than fully instrumented benchmark runs, and live empirical validation on real systems remains an essential and urgent next step. Additionally, the calibration head assumes access to the backbone model's internal hidden states; black-box API scenarios—where only output tokens are observable—require external uncertainty surrogates such as output distribution entropy or ensemble disagreement, which may be substantially less accurate than internal-state calibration.

VIII. Conclusion and Future Work

This paper has introduced CAAG, a calibrated behavioral control layer for agentic language models that makes act–ask–abstain decisions explicit, utility-grounded, and resource-bounded. The design is motivated by a precise diagnosis: existing frameworks leave the act/ask/abstain boundary implicit in prompting, which produces three structural failure modes—uncalibrated execution, over-tooling, and always-on reflection overhead—that are directly responsible for the large performance gaps observed across current agent benchmarks. CAAG addresses all three failure modes simultaneously through a two-layer architecture that attaches a lightweight calibration head to a frozen backbone and routes behavioral decisions through a utility-aware gating rule subject to formal risk and budget constraints. By treating basic agent behavior as a first-class optimization objective rather than an emergent property of prompting, CAAG provides a principled path to lower false-action rates, reduced tool-call overhead, and improved deployment suitability under computational constraints. Simulated evaluation projects substantial improvements across all six key metrics relative to existing frameworks, and ablation analysis confirms that each architectural component contributes independently and non-redundantly.

Future work should pursue four directions. First, live empirical validation on the full WebArena, SWE-bench, and VisualWebArena suites is essential to confirm the projected performance profile and to identify distributional shifts that the current analytical model does not capture. Second, multimodal extension of the calibration head to handle image, DOM, and text inputs jointly would extend CAAG's applicability to visually rich environments [13], where visual ambiguity is a major driver of false-action errors. Third, hierarchical gating—where a cheap local policy handles routine decisions and a more expensive global policy intervenes only on high-uncertainty or high-risk actions—could further reduce compu-

tational overhead on long-horizon tasks with many routine substeps. Fourth, online adaptation of calibration thresholds under non-stationary task distributions would address the threshold-tuning limitation identified in Section VII, enabling the policy to track distribution shifts that occur when agents are deployed on novel task categories. If the field is to close the gap between current agent performance and human-level reliability, behavioral calibration—not merely stronger generation—may be the most leveraged investment available to researchers and practitioners alike.

ACKNOWLEDGEMENT

The author sincerely thanks Dr. Payal Gulati for valuable mentorship, guidance, and continuous support throughout this research work. The author also expresses gratitude to Co-Mentor Piyush Gupta for technical insights, constructive feedback, and encouragement during the development of this study.

REFERENCES

[1] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, and Y. Cao, "ReAct: Synergizing Reasoning and Acting in Language Models," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2023. [Online]. Available: https://openreview.net/forum?id=WE_vluYUL-X

[2] T. Schick, J. Dwivedi-Yu, R. Dessi, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, and T. Scialom, "Toolformer: Language Models Can Teach Themselves to Use Tools," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, 2023.

[3] N. Shinn, F. Cassano, B. Labash, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language Agents with Verbal Reinforcement Learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, 2023. [arXiv:2303.11366].

[4] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. R. Narasimhan, "Tree of Thoughts: Deliberate Problem Solving with Large Language Models," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, 2023.

[5] Y. Geifman and R. El-Yaniv, "SelectiveNet: A Deep Neural Network with an Integrated Reject Option," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 2151–2159.

[6] H. Liu, Z.-Y. Dou, Y. Wang, N. Peng, and Y. Yue, "Uncertainty Calibration for Tool-Using Language Agents," in *Findings of the Assoc. Comput. Linguistics: EMNLP*, 2024, pp. 16781–16805.

[7] X. Liu *et al.*, "AgentBench: Evaluating LLMs as Agents," arXiv preprint arXiv:2308.03688, 2023.

[8] S. Zhou *et al.*, "WebArena: A Realistic Web Environment for Building Autonomous Agents," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2024. [arXiv:2307.13854].

[9] X. Deng, Y. Gu, B. Zheng, S. Chen, S. Stevens, B. Wang, H. Sun, and Y. Su, "Mind2Web: Towards a Generalist Agent for the Web," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, 2023. [arXiv:2306.06070].

[10] S. Yao, H. Chen, J. Yang, and K. R. Narasimhan, "WebShop: Towards Scalable Real-World Web Interaction with Grounded Language Agents," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2022. [arXiv:2207.01206].

[11] M. Shridhar, X. Yuan, M.-A. Coˆte´, Y. Bisk, A. Trischler, and M. Hausknecht, "ALFWorld: Aligning Text and Embodied Environments for Interactive Learning," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2021.

[12] E. Z. Liu, K. Guu, P. Pasupat, T. Shi, and P. Liang, "Reinforcement Learning on Web Interfaces Using Workflow-Guided Exploration," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2018.

[13] J. Y. Koh, R. Lo, L. Jang, V. Dua, Q. Zhong, R. Salakhutdinov, and D. Fried, "VisualWebArena: Evaluating Multimodal Agents on Realistic Visual Web Tasks," in *Proc. Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2024.

[14] H. Tao, S. T. Venkatesh, M. Shlapentokh-Rothman, and D. Hoiem, "WebWISE: Web Interface Control and Sequential Exploration with Large Language Models," arXiv preprint arXiv:2310.16042, 2023.

[15] Q. Wu *et al.*, "AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation," arXiv preprint arXiv:2308.08155, 2023.

[16] J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative Agents: Interactive Simulacra of Human Behavior," in *Proc. ACM Symp. User Interface Softw. Technol. (UIST)*, 2023.

[17] A. Madaan *et al.*, "Self-Refine: Iterative Refinement with Self-Feedback," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, 2023. [arXiv:2303.17651].

[18] P. Shaw, M. Joshi, J. Cohan, and K. Toutanova, "From Pixels to UI Actions: Learning to Follow Instructions via Graphical User Interfaces," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2023.

[19] R. Nakano *et al.*, "WebGPT: Browser-assisted Question-Answering with Human Feedback," arXiv preprint arXiv:2112.09332, 2021.

[20] C. E. Jimenez, J. Yang, A. Wettig, S. Yao, K. Pei, O. Press, and K. R. Narasimhan, "SWE-bench: Can Language Models Resolve Real-World GitHub Issues?" in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2024.

[21] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 1321–1330.

[22] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Aligning AI With Shared Human Values," arXiv preprint arXiv:2008.02275, 2020.

[23] X. Wang *et al.*, "Self-Consistency Improves Chain of Thought Reasoning in Language Models," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2023.

[24] L. Wang *et al.*, "A Survey on Large Language Model Based Autonomous Agents," *Frontiers Comput. Sci.*, vol. 18, no. 6, 2024, Art. no. 186345.

[25] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building Machines That Learn and Think Like People," *Behavioral Brain Sci.*, vol. 40, e253, 2017.