Adaptive Verification and Confidence-Aware Routing for Reliable Multi-Agent Large Language Model Reasoning

Andrew Hollis[1], Uday Naidu[2], and Darren Lowe[3]

[1]Department of Computer Science, University of Texas at Dallas, USA, andrew16@utdallas. edu
[2]Department of Computer Science, George Mason University, USA, unaidu@gmu. edu
[3]Department of Computer Science, University of Central Florida, USA, dl1984@ucf. edu

Abstract

Large-language-model (LLM) multi-agent systems can improve difficult reasoning by decomposing a problem, assigning specialized roles, and checking intermediate outputs. Most existing systems, however, invoke verification according to a fixed workflow and route subtasks using static role descriptions or uncalibrated self-reports. They consequently spend computation on low-risk states while allowing confidently wrong, high-impact states to propagate. We propose CAVR-MAS (Confidence-Aware Verification and Routing for Multi-Agent Systems), a control framework that couples feature-augmented confidence calibration, domain-conditioned reliability memory, uncertainty-aware routing, and multi-level verification. For every node in a subtask dependency graph, CAVR-MAS estimates calibrated correctness, path disagreement, downstream propagation risk, task criticality, and verification cost. A budget-adaptive controller selects among no check, critique, cross-agent checking, evidence-grounded checking, and re-decomposition by maximizing estimated error reduction per unit cost. Routing and final fusion use conservative posterior estimates of agent reliability rather than raw confidence or majority voting. We prove one-step Bayes optimality under correct estimators, a finite-error routing-regret bound, monotone verification workload with respect to the trigger threshold, a cumulative budget-violation bound, and consistency of nondiscounted reliability memory. We also define an executable evaluation protocol on GSM8K, MATH, PlanBench, MuSiQue, and HotpotQA, with controlled baselines, ablations, calibration analysis, and paired statistical tests. The result is a technically complete reliability-control formulation whose empirical claims are explicitly falsifiable.

Keywords: large language models; multi-agent systems; confidence calibration; uncertainty estimation; adaptive verification; dynamic routing; reliable artificial intelligence

1 Introduction

Chain-of-thought prompting, self-consistency, reasoning–action coupling, and structured search allocate additional inference-time computation to difficult problems and expose intermediate states that may be inspected or revised [1, 2, 3, 4]. Iterative self-feedback and verbal reflection further show that a generated trajectory need not be treated as final [5, 6]. These advances are important, but a single model remains vulnerable to correlated mistakes: an early assumption can condition all subsequent steps, and fluent self-critique can preserve rather than remove the original error.

Multi-agent LLM systems address part of this problem by distributing planning, generation, critique, and tool use across interacting roles. Debate elicits alternative arguments [7]; CAMEL, MetaGPT, AgentVerse, AutoGen, and ChatDev operationalize role-based collaboration or programmable conversation [8, 9, 10, 11, 16]. Yet adding agents is not itself a reliability mechanism. Unrestricted exchanges increase context length and can amplify correlated beliefs, whereas a fixed planner–executor pipeline may repeatedly invoke the same weak specialist or verifier.

A recent preprint by Wang, Feng, and Fang [17] represents the transition from unrestricted dialogue to structured collaboration. It separates planning, specialist reasoning, verification, and fusion; uses semantic communication and verifier-guided feedback; adapts agent weights; and studies communication-budget-aware execution. Its central systems insight is that routing, verification, memory organization, and communication structure matter as much as

agent count. The unresolved control question is more fine grained: *which intermediate states deserve verification, how intensive should the check be, and which agent should act next when confidence itself may be miscalibrated?* A verifier in the architecture does not answer when invoking that verifier has positive expected value.

This paper formulates verification and routing as coupled decisions under uncertainty and cost. Raw confidence is first transformed into an empirical probability of correctness using a held-out calibration set and auxiliary behavioral signals. The system then estimates the risk that a local error will affect descendants in the subtask graph. Verification is selective and graded rather than uniformly present or absent. Agent competence is maintained as a domain-specific posterior updated from audited outcomes, enabling the router to distinguish a semantically suitable agent from one that is reliably correct and calibrated in the relevant domain.

The work makes four methodological contributions. First, it introduces a confidence-aware routing rule that combines task–agent compatibility, calibrated correctness, conservative historical reliability, and inference cost. Second, it develops an adaptive verification controller that assigns one of five verification levels by estimating quality gain under a global resource constraint. Third, it defines a compact reliability memory that tracks domain-specific success, correction, cost, and calibration drift and is shared by routing and fusion. Fourth, it derives formal properties for routing, reliability learning, and resource control and specifies a falsifiable evaluation protocol that measures accuracy, calibration, error propagation, and computation jointly. The theory characterizes the controller under stated assumptions; benchmark superiority remains an empirical question rather than an architectural assertion.

2 Related Work

2.1 LLM Reasoning and Self-Correction

Chain-of-thought prompting elicits intermediate rationales, while self-consistency marginalizes over sampled reasoning paths [1, 2]. ReAct interleaves reasoning with actions in an external environment, and Tree of Thoughts searches over candidate states rather than committing to a single chain [3, 4]. Self-Refine and Reflexion use model-generated feedback or stored verbal experience to revise outputs [5, 6]. These methods provide useful uncertainty signals—path diversity, tool outcomes, critique content, and revision frequency—but typically do not calibrate those signals into a task-conditional probability of correctness used for cross-agent control.

2.2 Multi-Agent LLM Collaboration

Multiagent debate seeks convergence through iterative argument exchange [7]. CAMEL studies role-conditioned communicative agents, MetaGPT encodes standardized operating procedures for software roles, AgentVerse organizes collaborative agent groups, AutoGen exposes programmable conversation patterns, and ChatDev applies communicative roles to software development [8, 9, 10, 11, 16]. Mixture-of-Agents instead aggregates heterogeneous model outputs through successive collaboration layers [12]. These frameworks establish that specialization and interaction topology affect performance. Their general-purpose interfaces, however, do not by themselves supply a calibrated policy for deciding when an additional message or agent call is warranted.

2.3 Structured Multi-Agent Collaboration and Verification

Process supervision evaluates intermediate reasoning rather than only final outcomes. Uesato et al. compare process- and outcome-based feedback, and Lightman et al. train step-level verifiers for mathematical reasoning [18, 19]. Such verifiers can localize errors, although their outputs may be biased, domain limited, or correlated with the generator.

Wang et al.'s 2026 preprint [17] is a particularly relevant structured multi-agent design: hierarchical decomposition is combined with specialist agents, a verifier, semantic message routing, adaptive weighting, fusion, and feedback iteration. AgentPrune addresses a complementary efficiency problem by pruning redundant messages from spatial–temporal communication graphs [13]. The present study adopts only the broad design principle that planning, execution, checking, and aggregation should have explicit responsibilities. It does not reproduce either architecture or its equations. The technical distinction is the control variable. Wang et al. primarily structure collaboration and reduce communication redundancy, while AgentPrune optimizes the communication graph; CAVR-MAS estimates calibrated node-level failure probability and uses it to allocate verification depth and select agents under a cost constraint. Verification is therefore an action chosen by estimated value of information, not a fixed role in every reasoning cycle.

Table 1: Conceptual comparison of representative approaches. A check indicates that the capability is an explicit method component, not merely implementable through a general framework. "Partial" denotes limited or task-specific support.

Method	Task decomposition	Agent specialization	Explicit verification	Dynamic routing	Confidence calibration	Adaptive verification	Cost awareness
Multiagent debate [7]	–	–	Partial	–	–	–	–
CAMEL [8]	Partial	✓	–	–	–	–	–
MetaGPT [9]	✓	✓	Partial	–	–	–	Partial
AgentVerse [10]	Partial	✓	Partial	Partial	–	–	–
AutoGen [11]	Config.	✓	Config.	Config.	–	–	Config.
Mixture-of-Agents [12]	–	✓	Partial	–	–	–	–
AgentPrune [13]	–	–	–	Partial	–	–	✓
RouteLLM [14]	–	✓	–	✓	–	–	✓
Wang et al. preprint [17]	✓	✓	✓	✓	–	Partial	✓
CAVR-MAS (proposed)	✓	✓	✓	✓	✓	✓	✓

2.4 Confidence Calibration and Uncertainty Estimation

Neural probabilities can be systematically miscalibrated, motivating post-hoc methods such as temperature scaling [20]. For LLMs, confidence may be obtained from token probabilities, verbalized probabilities, repeated samples, or semantic equivalence classes. Tian et al. show that elicitation strategy materially affects calibration in instruction-following models [21], while semantic uncertainty groups meaning-equivalent generations before measuring entropy [22]. Surveys emphasize that confidence source, calibration method, and downstream action must be evaluated separately [23]. CAVR-MAS treats calibration as a supervised prediction problem and tests whether calibrated probabilities improve actions, not merely ECE.

2.5 Dynamic Routing and Cost-Aware Agent Systems

Routing among heterogeneous models or agents is related to selective prediction, cascades, and adaptive computation. RouteLLM learns cost-sensitive model selection from preference data, and unified cascade routing formalizes when sequential escalation and direct routing should be combined [14, 15]. These studies make estimated response quality central to economical inference, but they route whole queries among models rather than controlling verification inside a dependency graph. In multi-agent settings, the relevant statistics are additionally nonstationary because agent prompts, tools, and task mixtures can change. CAVR-MAS therefore uses a domain-conditioned reliability posterior, an explicit cost term, an online drift monitor, and downstream-risk features. This combination separates it from semantic-only role assignment and query-level model cascading.

Gap synthesis. Prior work provides strong components—structured roles, debate, process verification, semantic uncertainty, and cost-sensitive inference—but leaves their control signals weakly coupled. The missing object is a calibrated policy that jointly decides *who acts*, *whether the result is checked*, and *how much checking is justified* for each dependency-sensitive reasoning state.

3 Problem Formulation

Let an input task be $x \sim D$ with target answer $y^\star$. A planner maps x to a directed acyclic subtask graph $G = (V, E)$, where node $i \in V$ is a subtask and $(j, i) \in E$ means that i depends on the output of j. Let $A = \{1, \ldots, A\}$ denote heterogeneous agents and $L = \{0, 1, 2, 3, 4\}$ the available verification levels.

The state of node i at control round t is

$$s_i^t = (h_i, d_i, \mathrm{pa}(i), a_i^t, p_i^t, \hat{p}_i^t, u_i^t, v_i^t, b_i^t), \tag{1}$$

where h_i is a structured semantic representation, d_i is a domain/difficulty descriptor, $\mathrm{pa}(i)$ is the predecessor set, a_i^t is the assigned agent, p_i^t and $\hat{p}_i^t$ are raw and calibrated confidence, u_i^t is uncertainty, v_i^t is verification status, and b_i^t is remaining local budget. Agent a produces candidate $o_{i,a}$ with observed cost $c_{i,a}$.

A policy π selects both routing and verification actions. Its population objective is

$$\min_{\pi,\theta} \mathrm{E}_{(x,y^\star)\sim D}\left[\ell(\hat{y}, y^\star) + \lambda_c C_{\mathrm{tot}} + \lambda_e E_{\mathrm{prop}}\right] + \lambda_{\mathrm{cal}} L_{\mathrm{cal}}(\theta), \tag{2}$$

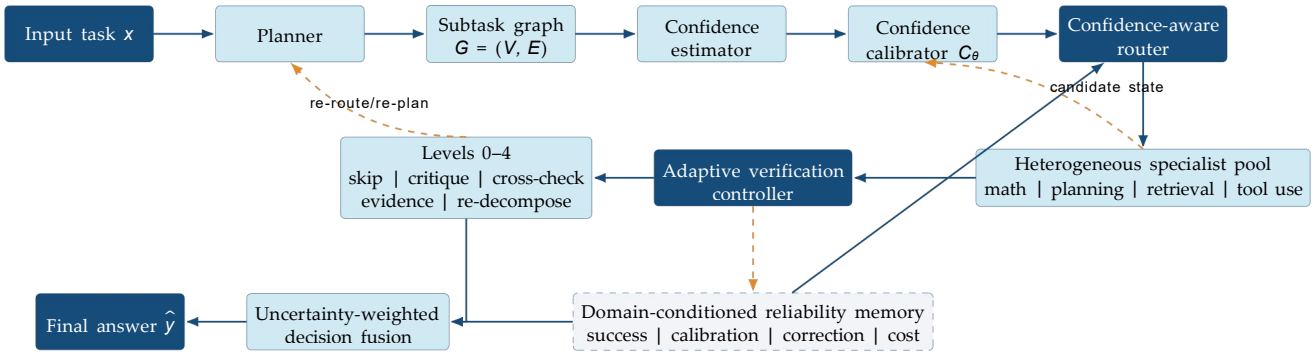

Figure 1: Original architecture of CAVR-MAS. Solid arrows denote task-state flow; dashed arrows denote control feedback and reliability updates. Unlike a fixed verifier stage, the controller may skip checking, increase checking depth, re-route, or return a node to the planner.

subject to $C_{\text{tot}} \le B$, a task-level budget. Here ℓ is final task loss, C_{tot} combines tokens, calls, latency, and monetary cost after normalization, E_{prop} counts incorrect nodes that influence an incorrect descendant, and L_{cal} is calibration loss. The formulation makes clear that high accuracy obtained by indiscriminate verification is not equivalent to an efficient policy.

4 Proposed Method

4.1 System Overview

CAVR-MAS executes a closed-loop controller over the subtask graph (figure 1). The planner creates nodes and dependencies; a confidence estimator extracts behavioral features; a post-hoc calibrator estimates correctness; the router chooses an agent using compatibility, reliability, uncertainty, and cost; and the verification controller selects verification intensity. Audited outcomes update reliability memory. Fusion aggregates only terminal candidates whose dependencies are resolved.

4.2 Task Decomposition

The planner emits for each node a schema $h_i = (q_i, O_i, C_i, \varepsilon_i)$ containing the local question, required output type, constraints, and admissible evidence/tools. Acyclicity is enforced by topological validation. Difficulty is estimated as $\kappa_i \in [0, 1]$ from prompt length, operator count, retrieval hops, tool requirements, and a learned planner score. If a decomposition is revised, descendants whose inputs changed are invalidated rather than silently reused.

4.3 Confidence Estimation and Calibration

For a candidate $o_{i,a}$, the estimator produces a raw correctness logit

$$g_{i,a} = \log \frac{p_{i,a} + \epsilon}{1 - p_{i,a} + \epsilon} \tag{3}$$

and auxiliary features $z_{i,a}$: normalized token entropy, agreement among K sampled paths, self-evaluation, evidence entailment, task difficulty, agent/domain identity, prior reliability, and verifier disagreement when available. Raw verbal confidence is retained as a feature but is never interpreted directly as a probability.

The primary calibration model is a feature-augmented Platt calibrator,

$$\hat{p}_{i,a} = C_\theta(p_{i,a}, z_{i,a}) = \text{sigmoid}\Big(\frac{g_{i,a}}{T} + \boldsymbol{w}^\top z_{i,a} + b\Big), \qquad T > 0. \tag{4}$$

Parameters $\theta = (T, \boldsymbol{w}, b)$ are fitted on a disjoint calibration split by minimizing binary NLL with L_2 regularization. This model is chosen because it preserves a transparent monotone dependence on raw confidence while allowing behavioral signals to correct domain and agent effects. Temperature-only, isotonic, and multilayer calibrators are sensitivity baselines. Calibration data are never reused as test data.

Node uncertainty is normalized Bernoulli entropy,

$$U_{i,a} = \frac{-\hat{p}_{i,a}\log\hat{p}_{i,a} - (1-\hat{p}_{i,a})\log(1-\hat{p}_{i,a})}{\log 2}. \tag{5}$$

Calibration is reported with ECE, Brier score, and NLL. Because ECE depends on binning, the main analysis uses equal-mass bins and reports adaptive-bin and classwise variants as robustness checks.

4.4 Uncertainty-Aware Verification Trigger

Suppose K candidate paths induce semantic answer classes with masses q_{ik}. Their disagreement is the normalized Gini impurity

$$D_i = \frac{1 - \sum_k q_{ik}^2}{1 - 1/K}, \tag{6}$$

with $D_i = 0$ for $K = 1$. Downstream propagation risk is

$$R_i = 1 - \prod_{j \in \mathrm{Desc}(i)} (1 - \omega_{ij}), \tag{7}$$

where ω_{ij} is the estimated probability that an error at i changes the feasible output at descendant j. Edge sensitivities are initialized from structural rules and updated from counterfactual repla y on development data.

Criticality $K_i \in [0, 1]$ measures whether the node lies on all feasible answer paths, while $\bar{C}_i$ is predicted verification cost normalized within the task.

After standardizing components on the calibration split, priority is

$$P_i = \mathrm{sigmoid}\left(\alpha\tilde{U}_i + \beta\tilde{D}_i + \gamma\tilde{R}_i + \delta\tilde{K}_i - \lambda\tilde{C}_i + b_v\right). \tag{8}$$

Verification is considered when $P_i > \tau_t$. The threshold is budget adaptive:

$$\tau_{t+1} = \Pi_{[\tau_{\min},\tau_{\max}]}\left(\tau_t + \eta_\tau \frac{C_{1:t} - B_{1:t}}{B + \epsilon}\right), \tag{9}$$

so overspending raises the bar for subsequent checks. A high-confidence node can still be verified if it is critical and has high propagation risk; an uncertain leaf can be skipped when its answer is low impact and checking is expensive.

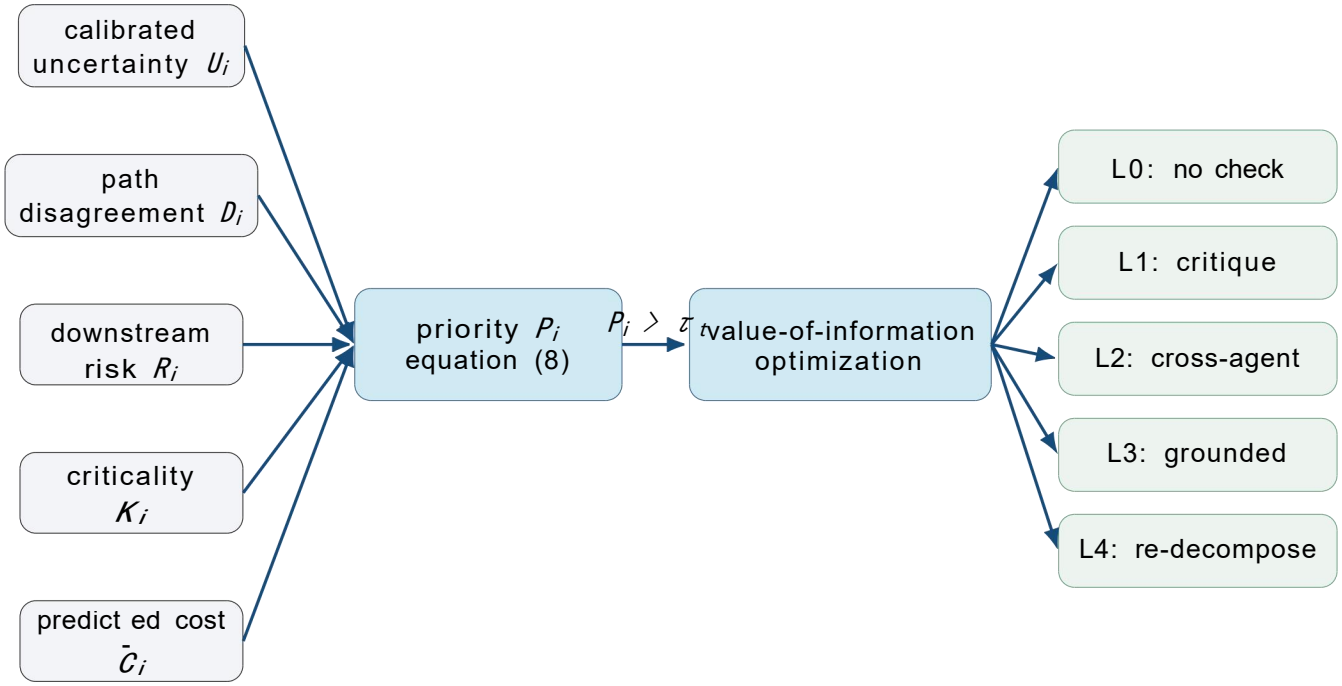

Figure 2: Adaptive verification. The selected level is based on expected error reduction net of cost, not on uncertainty alone.

4.5 Adaptive Verification Intensity

The action set is: level 0, accept without an additional call; level 1, one independent critique; level 2, two independent cross-agent checks followed by adjudication; level 3, verification against executable or retrieved evidence; and level 4, re-decomposition followed by re-routing. Not every task supports evidence-grounded checking, so infeasible actions are masked.

Let $\Delta Q_{i,l}$ be the reduction in expected node error after verification level l, estimated by a development-set benefit model $f_\phi(s_i, l)$. The controller chooses

$$l_i^* = \arg\max_{l \in \mathcal{L}_i} \left[f_\phi(s_i, l) - \mu_t \widehat{\mathrm{Cost}}(i, l) - \chi \, \mathrm{Var}\{\Delta Q_{i,l}\} \right], \tag{10}$$

where μ_t is a dual cost multiplier and χ penalizes uncertain benefit estimates. If all nonzero levels have nonpositive net value, $l_i^* = 0$. Let c_t be realized control cost at round t and let b_t be its allocated budget, with $\sum_{t=1}^{T} b_t = B$. The multiplier is updated by projected dual ascent,

$$\mu_{t+1} = [\mu_t + \eta_\mu (c_t - b_t)]_+ . \tag{11}$$

The benefit model is trained only on development trajectories generated by randomized verification assignments, reducing selection bias between easy nodes and cheap actions.

4.6 Confidence-Aware Agent Routing

For node i and agent a, the router computes

$$S_{i,a} = \theta_m M_{i,a} + \theta_h H_{a,d_i} + \theta_q \widehat{Q}_{i,a} - \theta_c \widehat{C}_{i,a} - \theta_u \widehat{U}_{i,a}, \tag{12}$$

where $M_{i,a}$ is semantic task–agent compatibility, H_{a,d_i} is a conservative reliability estimate, $\widehat{Q}_{i,a}$ predicts correctness after calibration, and $\widehat{C}_{i,a}$ and $\widehat{U}_{i,a}$ are predicted cost and uncertainty. During evaluation the deterministic policy uses $a_i^* = \arg\max_a S_{i,a}$. During development, a low-temperature softmax supplies exploration and prevents an initially favored agent from monopolizing feedback.

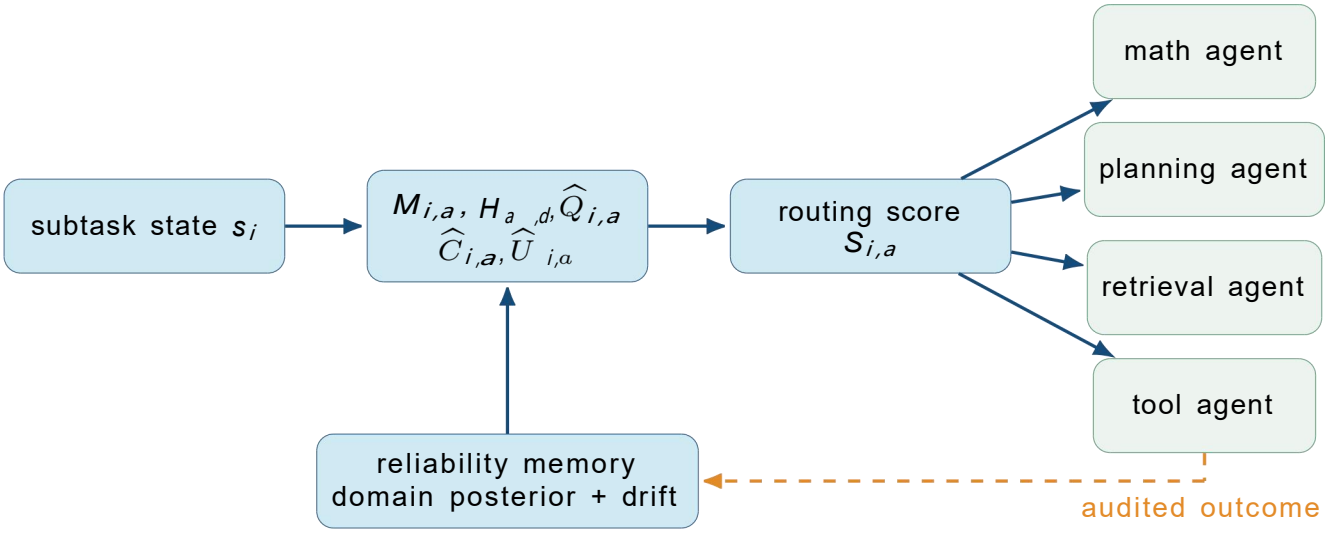

Figure 3: Confidence-aware routing among heterogeneous agents. Reliability is domain conditioned and updated only from auditable outcomes.

4.7 Reliability Memory

For agent a and domain d, correctness is represented by a discounted Beta evidence state with parameters $(\alpha_{a,d}^t, \beta_{a,d}^t)$:

$$\alpha_{a,d}^{t+1} = \varphi \alpha_{a,d}^t + w_t r_t, \tag{13}$$

$$\beta_{a,d}^{t+1} = \varphi \beta_{a,d}^t + w_t (1 - r_t), \tag{14}$$

initialized at (α_0, β_0), where $r_t \in \{0, 1\}$ is an audited outcome, $w_t \in [0, 1]$ reflects audit quality, and $\varphi \in (0, 1]$ is a forgetting factor. For $\varphi = 1$ and $w_t = 1$, this is the ordinary Beta–Bernoulli posterior; $\varphi < 1$ discounts stale evidence under drift. Writing $\Theta^t_{a,d} \sim \mathrm{Beta}(\alpha^t_{a,d}, \beta^t_{a,d})$, the conservative routing statistic is

$$H^t_{a,d} = \mathbb{E}[\Theta^t_{a,d}] - \kappa_h \sqrt{\mathrm{Var}(\Theta^t_{a,d})} - \xi\, \mathrm{ECE}^t_{a,d}\,. \tag{15}$$

Memory also stores exponentially weighted cost, verifier acceptance, correction frequency, and sample count. A two-window test on Brier score and accuracy flags calibration drift; flagged domains trigger recalibration or shrinkage toward the global prior. Verifier acceptance alone is not treated as ground truth because verifier and generator errors may be correlated.

4.8 Decision Fusion

Let Y be semantic equivalence classes of terminal candidate answers. Each candidate $o_{i,a}$ contributes log evidence to class y:

$$L(y) = \log \pi(y) + \sum_{(i,a):o_{i,a} \mapsto y} \omega_{i,a} \log \frac{\hat{p}_{i,a}}{1 - \hat{p}_{i,a}} + \nu_v V_{i,a} + \nu_e E_{i,a}, \tag{16}$$

where $V_{i,a}$ is verifier support, $E_{i,a}$ is evidence consistency, and

$$\omega_{i,a} = \frac{H_{a,d_i}}{1 + \zeta\ \mathrm{corr}_a(i)} \tag{17}$$

discounts correlated candidates from the same agent or shared ancestry. The answer is $\hat{y} = \arg\max_y L(y)$, and fused confidence is obtained by softmax normalization followed by a second held-out temperature scaling step. This prevents a large number of dependent votes from creating spurious certainty.

4.9 Algorithm

Algorithm 1 Confidence-Aware Adaptive Verification and Routing

Require: task x, agents A, budget B, calibrator C_θ, reliability memory M
Ensure: final answer $\hat{y}$

$G \leftarrow \mathrm{DECOMPOSE}(x)$; validate dependencies and node schemas
for i in a topological work queue of G **do**
  form state s_i from predecessors and remaining budget
  for $a \in A$ feasible for i **do**
    predict match, quality, uncertainty, and cost; read H_{a,d_i} from M
    compute $S_{i,a}$ by equation (12)
  end for
  $a_i \leftarrow \arg\max_a S_{i,a}$; $o_{i,a_i}, p_{i,a_i} \leftarrow \mathrm{EXECUTE}(a_i, s_i)$
  extract z_{i,a_i} and calibrate $\hat{p}_{i,a_i} \leftarrow C_\theta(p_{i,a_i}, z_{i,a_i})$
  compute $U_i, D_i, R_i, K_i, \bar{C}_i$ and priority P_i by equation (8)
  if $P_i > \tau_t$ **then**
    $l_i \leftarrow \arg\max_{l \in L_i} f_\phi(s_i, l) - \mu_t \widehat{\mathrm{Cost}}(i, l) - \chi \widehat{\mathrm{Var}}(i, l)$
  else
    $l_i \leftarrow 0$
  end if
  $o_i', v_i \leftarrow \mathrm{VERIFY}(o_{i,a_i}, l_i)$
  if $l_i = 4$ or v_i requests structural revision **then**
    $G \leftarrow \mathrm{REDECOMPOSEANDINVALIDATEDESCENDANTS}(G, i)$; continue
  else if v_i requests another specialist **then**
    exclude failed route and return i to work queue
  else
    commit o_i' and release newly ready descendants
  end if
  update M, τ_t, and μ_t using audited outcome and realized cost
end for
$y \leftarrow \mathrm{FUSE}(\text{terminal candidates}, M)$ using equation (16)
return $\hat{y}$

4.10 Computational Complexity

Let $n = |V|$, A be the number of candidate agents, K the number of sampled paths, and $m_l(l)$ the calls made by verification level l. Feature-based routing costs $O(nA)$ score evaluations, and disagreement estimation costs $O(nK)$ plus semantic clustering. LLM inference dominates, with total calls

$$N_{\text{call}} = n + \sum_{i=1}^{n} m_i(l_i) + N_{\text{replan}}. \tag{18}$$

Uniform level-L verification instead incurs $n[1 + m(L)]$. CAVR-MAS is computationally advantageous only if selective reductions in verification calls and propagated errors outweigh calibration, routing, and memory overhead; this condition is measured rather than assumed.

5 Experimental Setup

5.1 Research Questions and Hypotheses

RQ1 asks whether confidence-aware verification improves task accuracy and long-chain reliability relative to fixed verification. **RQ2** asks whether calibrated confidence improves agent selection relative to raw self-confidence. **RQ3** asks whether adaptive verification improves the accuracy–cost Pareto frontier. **RQ4** asks how routing affects error propagation through the subtask graph. **RQ5** asks which components determine the reliability–efficiency trade-off.

The corresponding prespecified hypotheses are: **H1**, selective graded verification reduces long-chain error relative to uniform verification at matched cost; **H2**, calibrated confidence lowers routing regret relative to raw confidence; **H3**, CAVR-MAS weakly dominates always-verify policies on the empirical accuracy–cost frontier; and **H4**, domain-specific reliability memory improves selection under heterogeneous task mixtures. Failure to reject a null hypothesis is reported without reinterpretation as support.

5.2 Datasets

The evaluation spans arithmetic reasoning (GSM8K [24]), competition mathematics (MATH [25]), classical planning (PlanBench [26]), compositional multi-hop QA (MuSiQue [27]), and retrieval-based multi-hop QA (HotpotQA [28]). Official test sets, licenses, answer normalization, and evaluation scripts will be used where released. A fixed calibration subset is sampled from training/development data before prompt or threshold tuning.

5.3 Baselines

The executable baseline set is: direct single-LLM answering; chain-of-thought; self-consistency at matched sample count; ReAct when tools are available; multiagent debate; a fixed planner–executor system; a uniform-verifier system; confidence-aware routing without adaptive verification; adaptive verification without calibration; and full CAVR-MAS. Recognized frameworks such as AutoGen or MetaGPT are reported as implementation baselines only if their public code, prompts, and compatible model interface are actually executed. Conceptual inclusion in table 1 is not an experimental claim.

Budget matching is performed in two ways: equal maximum token budget and equal realized monetary cost within a tolerance fixed before evaluation. For debate and self-consistency, sample count is adjusted only on the development set. The same backbone model, tool endpoints, context limit, and answer parser are used whenever method definitions permit.

5.4 Implementation and Reproducibility Protocol

The following fields must be frozen in the experiment manifest before test evaluation:

- backbone model provider, exact model/version identifier, access date, tokenizer, API parameters, and whether log probabilities are available;
- complete system and role prompts, few-shot exemplars, maximum output tokens, temperature, top-p, stop conditions, retry policy, and tool descriptions;

Table 2: Benchmark portfolio and evaluation roles specified by the empirical protocol.

Dataset	Category	Primary challenge	Primary metric	Calibration split	Tools/evidence
GSM8K	Mathematics	multi-step grade-school arithmetic	exact match	fixed train subset	optional calculator
MATH	Mathematics	competition-level symbolic reasoning	exact match	fixed train subset	optional Python/CAS
PlanBench	Planning	executable action sequence and state change	plan validity	development domains	symbolic validator
MuSiQue	Multi-hop QA	compositional evidence integration	EM/F1	development subset	supplied passages
HotpotQA	Multi-hop QA	retrieval and supporting-fact reasoning	EM/F1	development subset	retrieval corpus

- agent pool composition, verifier models, semantic clustering model, maximum graph size, maximum communication rounds, and re-decomposition limit;
- calibration split identifiers, optimizer, regularization, ECE binning rule, threshold range, cost weights, and all routing coefficients;
- dataset commit/version, preprocessing hashes, official scoring scripts, random seeds, machine type, GPU/CPU model, memory, software environment, concurrency, and measured wall-clock policy.

Primary experiments use one frozen commercial or open-weight backbone across roles; a second-backbone replication tests model dependence. If an API model changes during collection, pre- and post-change results are not pooled. Prompts, anonymized traces permitted by dataset and provider terms, calibration parameters, and evaluation code will accompany submission.

5.5 Evaluation Metrics

Task metrics include accuracy or completion rate, exact match/F1 where appropriate, and valid-plan rate. Long-chain reasoning error rate is

$$\mathrm{LCER} = \frac{\#\{\text{failed tasks with an incorrect ancestor state}\}}{\#\{\text{evaluated tasks with at least two dependent states}\}}. \tag{19}$$

Reliability metrics are ECE, Brier score, NLL, verifier agreement, correction success, and cross-agent decision consistency. Efficiency metrics are input/output tokens, model calls, verifier calls, communication rounds, latency, peak context, and estimated monetary cost. Joint utility is reported only as a sensitivity family,

$$\mathrm{Utility}(\lambda) = \mathrm{Accuracy} - \lambda \widetilde{\mathrm{Cost}}, \tag{20}$$

rather than for a single arbitrary λ. Routing regret is the correctness/cost gap between the selected agent and the best retrospectively observed feasible agent on development audit tasks.

5.6 Statistical Protocol

Stochastic methods are run with at least five prespecified seeds. For paired benchmark items, accuracy differences use paired bootstrap 95% confidence intervals and McNemar's test when its assumptions hold. Continuous per-item costs use paired bootstrap intervals and a permutation test; standardized paired effect sizes are also reported. Familywise error across the five primary research questions is controlled by Holm's procedure. Deterministic outputs from a single frozen run are not assigned artificial run-level standard deviations; uncertainty is instead estimated over benchmark items. Pareto dominance is assessed with bootstrap confidence bands. All exclusions, parser failures, and API retries are reported.

6 Analytical Properties and Evaluation Commitments

This section states what follows from the formulation itself and separates those properties from claims that require benchmark execution. The analytical statements concern decision consistency and resource control; they do not imply that the learned estimates are accurate in a particular deployment.

6.1 One-Step Optimality of Cost-Aware Routing

Define the conditional utility of assigning agent a to node i as

$$J_{i,a} = \Pr(o_{i,a} \text{ is correct} \mid s_i, a) - \lambda_c \mathbb{E}[c_{i,a} \mid s_i, a]. \tag{21}$$

Suppose $\widehat{Q}_{i,a}$ and $\widehat{C}_{i,a}$ equal the two conditional expectations above, and absorb semantic compatibility and domain reliability into $\widehat{Q}_{i,a}$. Then equation (12) with $\theta_q = 1$, $\theta_c = \lambda_c$, and $\theta_m = \theta_h = \theta_u = 0$ selects $\arg\max_a J_{i,a}$.

Proposition 1 *Under these assumptions, deterministic routing by equation* (12) *minimizes one-step Bayes risk among the feasible agents*

Proof. The one-step loss is $1 - r_{i,a} + \lambda_c c_{i,a}$, where $r_{i,a}$ is the correctness indicator. Conditional expectation gives $1 - J_{i,a}$. Minimizing this quantity is equivalent to maximizing $J_{i,a}$, which is precisely the routing decision. □

The exact-estimator assumption can be relaxed. Let $Q_{i,a} = \Pr(o_{i,a} \text{ is correct} \mid s_i, a)$ and $C_{i,a} = \mathrm{E}[c_{i,a} \mid s_i, a]$, and define $\widehat{J}_{i,a} = \widehat{Q}_{i,a} - \lambda_c \widehat{C}_{i,a}$.

Proposition 2 *If $|\widehat{Q}_{i,a} - Q_{i,a}| \le \epsilon_Q$ and $|\widehat{C}_{i,a} - C_{i,a}| \le \epsilon_C$ uniformly over feasible agents, then the utility regret of $\hat{a} = \arg\max_a \widehat{J}_{i,a}$ satisfies*

$$J_{i,a^\star} - J_{i,\hat{a}} \le 2(\epsilon_Q + \lambda_c \epsilon_C), \tag{22}$$

where $a^\star = \arg\max_a J_{i,a}$.

Proof. Let $\epsilon = \epsilon_Q + \lambda_c \epsilon_C$. Uniform estimation error gives $|\widehat{J}_{i,a} - J_{i,a}| \le \epsilon$. Hence $J_{i,a^\star} \le \widehat{J}_{i,a^\star} + \epsilon \le \widehat{J}_{i,\hat{a}} + \epsilon \le J_{i,\hat{a}} + 2\epsilon$. □

These propositions are deliberately local. Globally optimal routing may differ because an agent's output changes descendant states and future verification costs. The propagation-risk and criticality terms are practical approximations to that sequential dependence.

Table 3: Executable contracts for the principal comparison systems. Each row fixes the intervention rather than naming a loosely specified prompting style.

Method	Paths	Planner	Verification policy	Routing policy	Budget control
Direct LLM	1	none	none	single model	one generation cap
Chain-of-thought	1	implicit	none	single model	matched token cap
Self-consistency	K	implicit	final-answer vote	single model	matched total calls
ReAct	1	implicit	tool observations	tool-enabled model	matched tool/call cap
Multiagent debate	A	none	peer critique each round	fixed participants	fixed rounds
Planner–executor	1/node	explicit	none	static role match	graph/call cap
Uniform verifier	1/node	explicit	same check at every node	static role match	matched realized cost
CAVR-MAS	adaptive	explicit	levels 0–4 by equation (10)	score in equation (12)	primal–dual controller

6.2 Monotone Verification Workload

For fixed priorities $P_1, \ldots, P_n$, define the trigger set $T(\tau) = \{i : P_i > \tau\}$. If $\tau_2 \ge \tau_1$, then $T(\tau_2) \subseteq T(\tau_1)$.

Proposition 3. *If all nonzero verification levels have nonnegative cost and level selection for retained nodes is unchanged, total verification cost is monotone nonincreasing in τ.*

Proof. Increasing the threshold can only remove nodes from the trigger set. The cost sum therefore loses nonnegative terms and cannot increase. □

This property establishes a predictable resource-control direction, not monotonic accuracy. Excessively large thresholds may suppress valuable checks, whereas very small thresholds can waste budget and delay critical nodes.

Proposition 4. *For the dual update in equation* (11)*, cumulative budget violation after T rounds satisfies*

$$\sum_{t=1}^{T} (c_t - b_t) \le \frac{\mu_{T+1} - \mu_1}{\eta_\mu}. \tag{23}$$

Consequently, if $\mu_1 = 0$ and $\mu_t \le \mu_{\max}$, violation is at most $\mu_{\max}/\eta_\mu$.

Proof. Projection onto the nonnegative half-line implies $\mu_{t+1} \ge \mu_t + \eta_\mu(c_t - b_t)$. Summing over t and rearranging yields equation (23). □

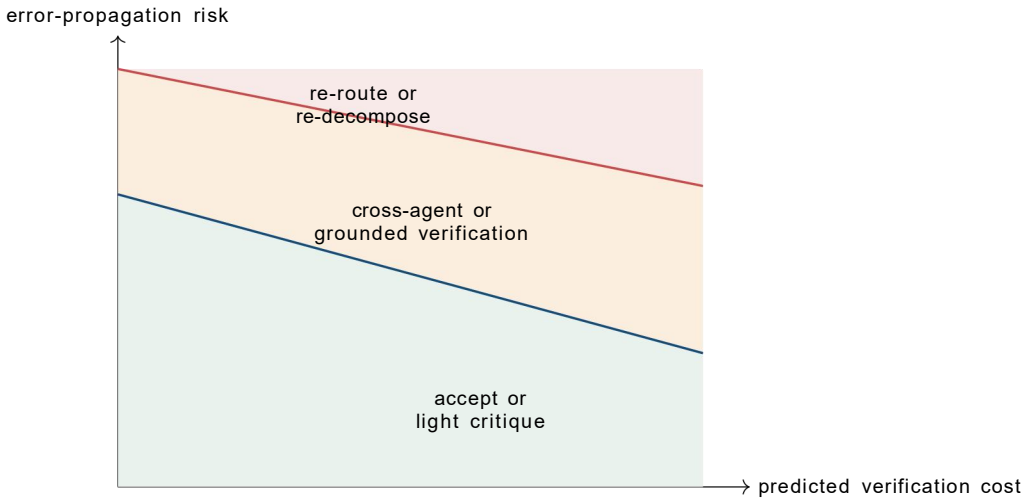

Figure 4: Decision geometry induced by risk and cost. The boundaries are conceptual level sets of equation (10), not empirical performance curves.

6.3 Consistency of Reliability Memory

The nondiscounted memory has a standard consistency guarantee that clarifies the role of audited feedback.

Proposition 5. *Fix an agent–domain pair and assume* $\varphi = 1$, $w_t = 1$, *and independent audited outcomes* $r_t \sim \text{Bernoulli}(q)$. *Then the posterior mean*

$$\bar{\Theta}_t = \frac{\alpha_0 + \sum_{s=1}^{t} r_s}{\alpha_0 + \beta_0 + t} \tag{24}$$

converges almost surely to q, *and the Beta posterior variance converges to zero.*

Proof. The strong law of large numbers gives $t^{-1} \sum_{s=1}^{t} r_s \to q$ almost surely. Dividing numerator and denominator of Θ_t by t proves mean convergence. The Beta variance is $\alpha_t\beta_t/[(\alpha_t + \beta_t)^2(\alpha_t + \beta_t + 1)]$, which is $O(t^{-1})$. □

For $\varphi < 1$, the memory deliberately sacrifices consistency under stationarity to track recent competence. This bias–adaptivity trade-off is measured by varying φ and by reporting both pre- and post-drift calibration.

6.4 Calibration as an Operational Control Signal

Calibration is useful only when it changes actions in the correct direction. Let $e_{i,a} = \mathbb{I}[o_{i,a} \neq y_i^\star]$. A calibrated score satisfies $\mathbb{E}[e_{i,a} \mid \hat{p}_{i,a} = q] = 1 - q$ up to finite-sample error. Substitution into equation (8) makes verification probability responsive to empirical error frequency rather than verbal certainty. The calibration study therefore reports ECE, Brier score, NLL, error-detection AUROC, and routing regret; no single metric is treated as sufficient.

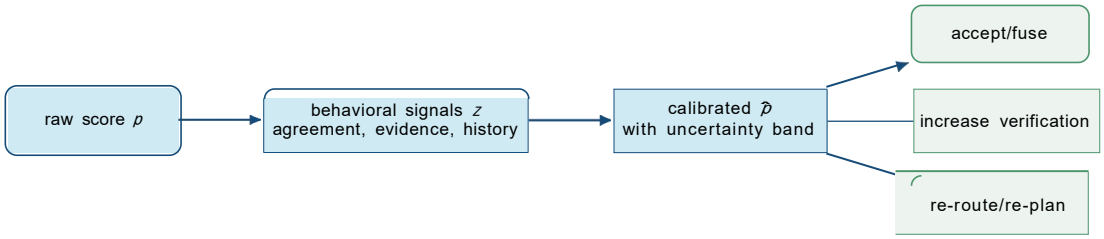

Figure 5: Operational role of calibration. The calibrated probability controls acceptance, verification, and routing decisions; calibration is not an end metric.

6.5 Ablation Logic

Table 4: Controlled ablation design and its causal interpretation.

Variant	Controlled replacement	Mechanism isolated	Primary readout
w/o calibration	use raw $p_{i,a}$ everywhere	probability calibration	ECE; routing regret
temperature-only	set $\boldsymbol{w} = 0$ in equation (4)	auxiliary behavioral features	Brier; LCER
w/o reliability memory	replace $H_{a,d}$ by a global constant	historical domain competence	selection accuracy
w/o adaptive trigger	verify every committed node	selective verification	calls at matched accuracy
w/o level selection	use level 1 whenever triggered	graded verification value	correction success
w/o uncertainty routing	set $\theta_u = 0$	anticipatory uncertainty penalty	routing regret
w/o cost penalty	set $\theta_c = \lambda = 0$	resource awareness	tokens; latency; cost
random routing	sample a feasible agent uniformly	learned allocation	task accuracy
fixed verification	use one prespecified level	value-of-information policy	Pareto dominance

H2 is evaluated by the calibrated-versus-raw contrast at the node level and by final task outcomes. H4 is evaluated both on the original mixture and after a controlled shift in domain proportions. The same cached candidate generations are reused where an intervention does not alter upstream context, reducing stochastic and financial variance without leaking test labels.

6.6 Efficiency Accounting and Threshold Sensitivity

Table 5: Efficiency-accounting rules used for every system.

Measure	Included operations	Aggregation
Tokens	prompts, outputs, tool wrappers, summaries	per task and total
Model calls	successful calls and billed retries	mean, median, tail
Verifier calls	critique, adjudication, evidence checks	per node and task
Latency	queue, inference, tools, orchestration	wall clock and critical path
Monetary cost	provider charges at logged access date	per solved task
Peak context	largest serialized agent state	tokens

Threshold τ, cost multiplier λ, number of agents, maximum messages, verification calls, calibration temperature, and forgetting factor φ are varied around the development-selected configuration. The exact monotonic statement is limited to workload for fixed priorities (Proposition 3); accuracy and calibration remain empirical quantities.

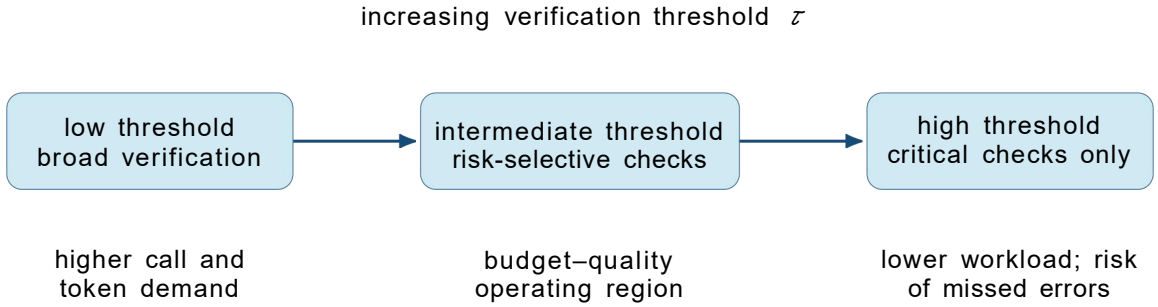

Figure 6: Analytical threshold response. Verification workload decreases with τ for fixed priorities, while accuracy must be established empirically.

6.7 Statistical Decision Rules

Table 6: Prespecified inferential decisions for the central research questions.

Question	Primary contrast	Test/unit	Decision criterion
RQ1	full vs. uniform verifier	paired bootstrap/item	LCER CI excludes zero at matched cost
RQ2	calibrated vs. raw routing	permutation/node	lower routing regret after Holm correction
RQ3	full vs. always verify	bootstrap/task	nondominated accuracy–cost region
RQ4	memory vs. no memory	paired bootstrap/task	benefit persists under domain shift
RQ5	complete ablation family	hierarchical contrasts	efect sizes with adjusted CIs

7 Discussion

The framework tests a stronger claim than “verification helps.” Its relevant hypothesis is that verification has heterogeneous marginal value across reasoning states and that calibrated uncertainty, disagreement, graph risk, and cost can predict this value. An accuracy gain at equal cost would support selective allocation; fewer calls accompanied by lower accuracy would characterize the controller as an efficiency heuristic rather than a reliability method. Calibration improvement without routing improvement would show that confidence is descriptively calibrated but operationally insufficient.

This framing extends the structured-collaboration hypothesis advanced by Wang et al. [17]. Their preprint argues that role separation, semantic communication, verifier feedback, and budget-aware coordination are preferable to unrestricted conversation. CAVR-MAS asks the next systems question: whether the same structured pipeline becomes more efficient when communication and verification are allocated according to calibrated risk instead of uniformly. Evidence for that statement requires the matched-budget comparisons, ablations, and Pareto analysis specified above; it cannot be inferred from architectural plausibility.

The specified contrasts admit informative null findings. A negligible reliability-memory effect under domain shift would suggest that agent errors are too correlated or that a shared backbone limits meaningful specialization. Frequent level-4 re-decomposition would expose a weak planner or a poorly calibrated benefit model. Disagreement may outperform self-reported confidence for some models, so feature analysis must inspect coefficients and conditional performance rather than assuming that verbal confidence is the principal signal.

The framework also distinguishes prediction from intervention. A controller can accurately forecast failure yet choose an ineffective correction, especially when the verifier shares the generator’s blind spot. Accordingly, verification-level benefit is learned from observed corrections, and the paper reports correction success in addition to error-detection AUROC. This separation is essential for reliable-agent evaluation.

8 Limitations

First, calibration is conditional on a backbone, prompt family, domain mixture, and time period. API updates or distribution shift can invalidate C_θ, and the drift detector reacts only after enough audited outcomes exist. Second, confidence features may be unavailable for closed models without token log probabilities; verbal confidence and sample agreement are imperfect substitutes.

Third, verifiers are not independent oracles. Agents sharing a backbone, retrieval corpus, or prompt may make correlated errors, causing both reliability memory and fusion to be overconfident. Evidence-grounded verification is also limited by tool correctness and retrieval coverage. Fourth, decomposed benchmarks may not represent open-ended scientific, social, or safety-critical tasks, and exact-match correctness does not measure all aspects of reasoning quality.

Fifth, adaptive control adds latency, implementation complexity, calibration data requirements, and attack surface. Reliability memory can encode historical dataset bias, while domain labels can be noisy. Sixth, the value-of-information model is trained from randomized development interventions; insufficient coverage of expensive verification levels will increase variance and may produce unstable policies. The analytical properties proved here concern local routing, estimator error, resource accounting, and stationary reliability learning; they do not establish benchmark superiority or global optimality of the full sequential policy.

9 Conclusion

This paper proposes CAVR-MAS, a multi-agent reasoning controller in which confidence calibration, routing, verification intensity, reliability memory, and fusion are coupled decisions. The method prioritizes intermediate states by uncertainty, disagreement, downstream risk, criticality, and cost; selects verification depth by estimated value of information; and routes subtasks using conservative domain-specific reliability. The design is distinct from fixed verifier pipelines because verification is an adaptive resource-allocation action rather than a mandatory stage. Five propositions characterize local routing optimality, finite estimator-error regret, monotone verification workload, cumulative budget violation, and reliability-memory consistency. The benchmark contracts, ablations, cost accounting, and statistical rules make the broader reliability claims directly falsifiable.

Data and Code Availability

No new dataset is introduced or analyzed in the theoretical development. The public benchmarks used by the evaluation protocol are cited at their original sources. All mathematical definitions and pseudocode required to implement the controller are contained in the manuscript; an empirical companion artifact must additionally archive prompts, configuration manifests, calibration-split identifiers, environment locks, eligible traces, and result-generation scripts.

Ethics Statement

The theoretical analysis uses no human participants, personal data, or newly collected human annotations. Any subsequent API- or benchmark-based evaluation must comply with the applicable model-provider terms, dataset licenses, privacy requirements, and restrictions on releasing generated traces.

Declaration of Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems*, vol. 35, pp. 24824–24837, 2022.

[2] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," in *International Conference on Learning Representations*, 2023.

[3] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing reasoning and acting in language models," in *International Conference on Learning Representations*, 2023.

[4] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan, "Tree of Thoughts: Deliberate problem solving with large language models," in *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[5] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegrefe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, et al., "Self-Refine: Iterative refinement with self-feedback," in *Advances in Neural Information Processing Systems*, vol. 36, pp. 46534–46594, 2023.

[6] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language agents with verbal reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[7] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch, "Improving factuality and reasoning in language models through multiagent debate," in *Proceedings of the 41st International Conference on Machine Learning*, PMLR 235, pp. 11733–11763, 2024.

[8] G. Li, H. A. A. K. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem, "CAMEL: Communicative agents for 'mind' exploration of large scale language model society," in *Advances in Neural Information Processing Systems*, vol. 36, pp. 51991–52008, 2023.

[9] S. Hong, M. Zhuge, J. Chen, X. Zheng, Y. Cheng, C. Zhang, J. Wang, Z. Wang, S. K. S. Yau, Z. Lin, et al., "MetaGPT: Meta programming for a multi-agent collaborative framework," in *International Conference on Learning Representations*, 2024.

[10] W. Chen, Y. Su, J. Zuo, C. Yang, C. Yuan, C. Qian, C.-M. Chan, Y. Qin, Y. Lu, R. Xie, et al., "AgentVerse: Facilitating multi-agent collaboration and exploring emergent behaviors," in *International Conference on Learning Representations*, 2024.

[11] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, et al., "AutoGen: Enabling next-gen LLM applications via multi-agent conversation," arXiv:2308.08155, 2023.

[12] J. Wang, J. Wang, B. Athiwaratkun, C. Zhang, and J. Zou, "Mixture-of-Agents enhances large language model capabilities," in *International Conference on Learning Representations*, 2025.

[13] G. Zhang, Y. Yue, Z. Li, S. Yun, G. Wan, K. Wang, D. Cheng, J. X. Yu, and T. Chen, "Cut the Crap: An economical communication pipeline for LLM-based multi-agent systems," in *International Conference on Learning Representations*, 2025.

[14] I. Ong, A. Almahairi, V. Wu, W.-L. Chiang, T. Wu, J. E. Gonzalez, M. W. Kadous, and I. Stoica, "RouteLLM: Learning to route LLMs from preference data," in *International Conference on Learning Representations*, 2025.

[15] J. Dekoninck, M. Baader, and M. Vechev, "A unified approach to routing and cascading for LLMs," in *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

[16] C. Qian, W. Liu, H. Liu, N. Chen, Y. Dang, J. Li, C. Yang, W. Chen, Y. Su, X. Cong, et al., "ChatDev: Communicative agents for software development," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 15174–15186, 2024.

[17] Wang, S., Feng, Y., & Fang, X. (2026, May). A Large Language Model-Enabled Multi-Agent Collaboration Method for Complex Task Solving. In 2026 6th International Symposium on Computer Technology and Information Science (ISCTIS) (pp. 253-256). IEEE.

[18] J. Uesato, N. Kushman, R. Kumar, F. Song, N. Siegel, L. Wang, A. Creswell, G. Irving, and I. Higgins, "Solving math word problems with process- and outcome-based feedback," arXiv:2211.14275, 2022.

[19] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe, "Let's verify step by step," in *International Conference on Learning Representations*, 2024.

[20] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning*, PMLR 70, pp. 1321–1330, 2017.

[21] K. Tian, E. Mitchell, A. Zhou, A. Sharma, R. Rafailov, H. Yao, C. Finn, and C. Manning, "Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback," in *Proceedings of EMNLP*, pp. 5433–5442, 2023, doi:10.18653/v1/2023.emnlp-main.330.

[22] L. Kuhn, Y. Gal, and S. Farquhar, "Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation," in *International Conference on Learning Representations*, 2023.

[23] J. Geng, F. Cai, Y. Wang, H. Koeppl, P. Nakov, and I. Gurevych, "A survey of confidence estimation and calibration in large language models," in *Proceedings of NAACL-HLT*, pp. 6577–6595, 2024.

[24] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, et al., "Training verifiers to solve math word problems," arXiv:2110.14168, 2021.

[25] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt, "Measuring mathematical problem solving with the MATH dataset," in *NeurIPS Datasets and Benchmarks Track*, 2021.

[26] K. Valmeekam, M. Marquez, A. Olmo, S. Sreedharan, and S. Kambhampati, "PlanBench: An extensible benchmark for evaluating large language models on planning and reasoning about change," in *Advances in Neural Information Processing Systems*, vol. 36, pp. 38975–38987, 2023.

[27] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "MuSiQue: Multihop questions via single-hop question composition," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 539–554, 2022, doi:10.1162/tacl__a__00475.

[28] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, "HotpotQA: A dataset for diverse, explainable multi-hop question answering," in *Proceedings of EMNLP*, pp. 2369–2380, 2018, doi:10.18653/v1/D18-1259.