Think Together, Think Smart: Adaptive Reasoning Budget Allocation for Multi-Agent Cooperative Systems

Anonymous CVPR submission

Paper ID ****

Abstract

Multi-agent systems increasingly rely on test-time reasoning, yet most existing pipelines allocate compute either uniformly across agents or independently without coordination. We present CoAR *(**Co**operative **A**daptive **R**easoning), a framework for reasoning-budget allocation in cooperative systems with interdependent sub-problems.* CoAR *has three stages: each agent first estimates local difficulty and selects a concise or deliberative reasoning mode; agents then emit confidence signals; finally, low-confidence outputs can trigger dependent agents to increase their reasoning budget and re-solve with shared context. We formalize two properties for cooperative reasoning:* cooperative monotonicity*, which measures whether harder composite tasks receive more total compute, and* cooperative compositionality*, which measures whether end-to-end reasoning cost is predictable from constituent sub-task costs. On a controlled two-agent compositional math benchmark, adaptive local budget allocation already improves Pass@1 from* 52.5% *to* 95.0% *while using roughly* $6\times$ *fewer tokens on easy instances than uniform slow reasoning. Our analysis further shows that confidence propagation is only as useful as the calibration of the uncertainty signal itself. These results suggest that in cooperative reasoning systems, adaptive allocation is the foundation, while calibrated uncertainty is the critical enabler for effective inter-agent coordination.*

1. Introduction

Cooperative intelligence emerges when multiple agents solve a composite task through structured dependencies. This setting appears in vehicle-to-vehicle perception, robot swarms, and LLM-based agent teams, where one agent's intermediate output often becomes another agent's input. In such systems, the central question is no longer only *what* to communicate, but also *how much reasoning compute* each agent should spend before and after communication.

Recent work on test-time scaling has shown that foundation models benefit from adaptive computation: difficult problems deserve longer reasoning traces, while easy problems can be solved concisely [11, 16, 21, 22]. However, these advances are largely single-agent. In cooperative settings, uncertainty can cascade across a dependency graph. An uncertain upstream output should influence not only the final answer, but also the reasoning budget of downstream peers.

We study this problem through CoAR, a simple framework for cooperative adaptive reasoning. Each agent first predicts the difficulty of its local sub-task and allocates either a FAST or SLOW budget. Agents then emit confidence estimates. If an upstream agent is sufficiently uncertain, its dependent peers increase their reasoning budget and re-solve with access to the uncertain output and its context. This mechanism resembles human teams: routine local steps are handled quickly, while uncertainty triggers broader, more careful coordination.

To analyze cooperative reasoning beyond raw accuracy, we introduce two system-level properties. **Cooperative monotonicity** asks whether more difficult composite tasks receive more total compute. **Cooperative compositionality** asks whether end-to-end cost can be predicted from the constituent sub-task costs with limited coordination overhead. Together, they provide a compact view of whether a system allocates reasoning effort in a principled way.

We evaluate CoAR on CoAR-MATH, a controlled benchmark of two-agent compositional math problems with varied difficulty asymmetry. Our findings are sharp. First, adaptive local allocation alone is highly effective: it outperforms both all-fast and all-slow baselines while preserving strong monotonicity. Second, explicit confidence propagation does not help unless uncertainty estimates are calibrated; in our experiments, self-reported confidence is often too optimistic to trigger re-reasoning when needed. These results suggest that adaptive allocation should be treated as the baseline capability for cooperative reasoning systems, and that calibrated uncertainty is the key bottleneck for richer coordination.

Our contributions are three-fold:

- We formulate cooperative reasoning-budget allocation for

multi-agent systems with task dependencies.

- We propose CoAR, a three-stage framework combining local adaptive reasoning, confidence propagation, and cooperative re-reasoning.
- We introduce cooperative monotonicity and cooperative compositionality as evaluation properties, and show empirically that confidence calibration—not the communication rule itself—is the main obstacle to effective cooperative adaptation.

2. Related Work

Cooperative multi-agent systems. Communication is a long-standing challenge in cooperative perception and multi-agent learning [2, 13, 14]. In autonomous driving, methods such as Where2comm and CoBEVT decide what perceptual information to transmit under bandwidth constraints [8, 18, 19]. Swarm robotics similarly studies scalable coordination under partial observability [4]. Our focus is complementary: instead of selecting sensory features to share, we study how agents should coordinate *reasoning effort* across interdependent sub-tasks.

LLM-based multi-agent reasoning. A growing literature uses multiple language agents for role specialization, debate, tool use, and collaborative planning [1, 5, 7, 10, 17]. These systems typically assume fixed prompting or fixed compute per agent. We ask a different systems question: when should one agent think longer because another agent is uncertain?

Adaptive test-time computation. Chain-of-thought prompting, tree search, chain-of-draft, and recent test-time scaling methods show that reasoning performance depends strongly on how much compute is allocated per input [3, 11, 15, 16, 20–23]. Prior work has also emphasized difficulty-aware scaling laws and structured transitions between fast and slow reasoning. Our work extends this line from single-agent to cooperative settings, where local difficulty is not the only signal that matters.

Confidence and calibration. Effective cooperative adaptation requires reliable uncertainty estimates. Prior work has shown that modern neural models can be miscalibrated [6], and LLM self-assessment is often imperfect [9]. This issue is central to our findings: propagation helps only when low-confidence events are detected faithfully.

Human team cognition. Dynamic reallocation of attention is a hallmark of human teams with shared mental models and mutual performance monitoring [12]. CoAR is inspired by this perspective: uncertainty should selectively recruit more downstream reasoning rather than induce uniform overhead everywhere.

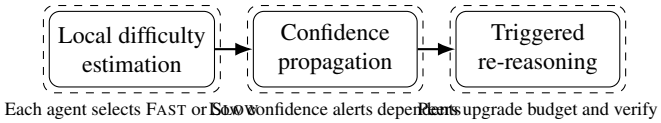

Figure 1. Overview of CoAR. Cooperative adaptation combines local difficulty-aware allocation with uncertainty-triggered downstream budget escalation.

3. Method

3.1. Problem Formulation

Consider N agents $\{a_1, \ldots, a_N\}$ solving a composite task T decomposed into sub-tasks $\{t_1, \ldots, t_N\}$. A directed dependency graph $G = (V, E)$ specifies information flow, where $(i, j) \in E$ means agent a_j depends on the output of agent a_i. Each agent produces an answer y_i and a reasoning trace r_i with token length $\ell(r_i)$. The total reasoning cost is

$$C(T) = \sum_{i=1}^{N} \ell(r_i). \tag{1}$$

We study the cooperative allocation problem

$$\arg\max_{\{B_i\}} \ \mathrm{Acc}(T) \quad \text{s.t.} \quad \arg\min \ C(T), \tag{2}$$

where B_i is the reasoning budget assigned to agent i under the dependency structure G.

3.2. Local Budget Allocation

Each agent estimates the difficulty of its local sub-task via a score $d(t_i) \in [1, 10]$. It then chooses between two discrete budgets,

$$B_i = \begin{cases} B_{\text{FAST}}, & d(t_i) \le \tau, \\ B_{\text{SLOW}}, & d(t_i) > \tau, \end{cases} \tag{3}$$

where τ is a difficulty threshold. In practice, B_{FAST} uses a concise direct-answer prompt and B_{SLOW} uses a more deliberative step-by-step prompt with a larger token budget.

3.3. Confidence Propagation

After the first-pass solution, each agent predicts a confidence score $c_i \in [0, 1]$. If $c_i < c_{\min}$, the agent broadcasts a low-confidence event to all dependent peers:

$$\text{Propagate}(i \to j) \iff (i, j) \in E \ \wedge \ c_i < c_{\min}. \tag{4}$$

This event carries the upstream output and the associated confidence signal. Importantly, propagation is selective: only uncertain states induce extra coordination cost.

3.4. Cooperative Re-Reasoning

When an agent a_j receives a propagation event from an upstream dependency, it upgrades its reasoning budget to B_{SLOW} regardless of its local difficulty estimate, verifies the

Algorithm 1 CoAR: Cooperative Adaptive Reasoning

Require: Agents $\{a_i\}_{i=1}^{N}$, dependency graph G, difficulty threshold τ, confidence threshold $c_{\min}$

for all agents a_i **in parallel do**
  $d(t_i) \leftarrow \text{EstimateDifficulty}(t_i)$
  $B_i \leftarrow B_{\text{FAST}}$ if $d(t_i) \leq \tau$ else B_{SLOW}
  $y_i, r_i \leftarrow \text{Solve}(t_i, B_i)$
  $c_i \leftarrow \text{AssessConfidence}(y_i, r_i)$
end for
for all agents a_i **do**
  if $c_i < c_{\min}$ **then**
    for all $(i, j) \in E$ **do**
      send (y_i, c_i) to a_j
    end for
  end if
end for
for all triggered agents a_j **do**
  upgrade $B_j \leftarrow B_{\text{SLOW}}$
  verify received upstream outputs
  $y_j \leftarrow \text{ReSolve}(t_j, B_j, \text{context})$
end for
return $\{y_i\}_{i=1}^{N}$

upstream answer, and re-solves its own sub-task conditioned on the received context. The mechanism is intentionally simple: uncertainty should move forward through the graph and trigger more careful downstream reasoning.

3.5. CoAR Protocol

Algorithm 1 summarizes the full procedure.

3.6. Properties of Cooperative Reasoning

We propose two properties for evaluating cooperative computation.

Cooperative monotonicity. Let aggregate task difficulty be

$$D(T) = \sum_{i=1}^{N} d(t_i). \tag{5}$$

A cooperative reasoning system is monotonic if harder composite tasks tend to receive at least as much total compute,

$$D(T_1) > D(T_2) \;\Rightarrow\; C(T_1) \geq C(T_2). \tag{6}$$

We quantify this behavior using the Spearman rank correlation $\rho(D, C)$.

Cooperative compositionality. Total reasoning cost should be predictable from the constituent sub-task costs. We measure this with normalized mean absolute deviation:

$$\text{nMAD}(T) = \frac{\left| C(T) - \sum_{i=1}^{N} C(t_i) \right|}{C(T)}. \tag{7}$$

Low nMAD indicates that coordination overhead remains structured rather than erratic.

4. Experiments

4.1. Setup

We construct CoAR-MATH, a benchmark of 40 two-agent cooperative math problems. Agent 1 solves a numerical sub-problem; Agent 2 solves a dependent sub-problem that uses Agent 1's answer. We design four balanced splits with ten instances each: **EE** (easy → easy), **EH** (easy → hard), **HE** (hard → easy), and **HH** (hard → hard). This controlled design isolates how methods respond to difficulty asymmetry across cooperating agents.

All experiments use Claude 3.5 Sonnet through the Anthropic API. We implement two prompt budgets: FAST uses a concise direct-answer prompt with `max_tokens`=200, and SLOW uses a deliberative prompt with `max_tokens`=2000. We compare four methods: **All-Fast**, **All-Slow**, **Adaptive-Indep**, and full CoAR with confidence propagation.

4.2. Metrics

We report: (i) **Pass@1**, where both agents must be correct; (ii) **Avg Tokens**, the total reasoning tokens across both agents; (iii) **REP**, a reasoning efficiency-performance score computed as

$$\text{REP} = \frac{\Delta\text{Acc}}{\Delta\text{Cost}_{\text{norm}}}, \tag{8}$$

where higher is better; and (iv) **Monotonicity** ρ, the Spearman rank correlation between $D(T)$ and $C(T)$.

4.3. Implementation Details

For each sub-task, the difficulty estimator is implemented by prompting the same backbone model to score difficulty on a 1–10 scale before solving. The final answer prompt is then switched between a short direct-response template and a longer step-by-step template according to the threshold τ. Confidence is obtained from a separate self-assessment prompt after generation. This setup intentionally keeps the adaptation mechanism lightweight and model-agnostic: no extra verifier, no fine-tuning, and no learned communication policy are required. The resulting benchmark therefore isolates the systems question of reasoning-budget allocation from confounding architectural changes.

4.4. Main Results

Table 1 shows that adaptive local allocation is already highly effective. **Adaptive-Indep** achieves the best accuracy at 95.0%, improving substantially over both **All-Fast** (52.5%) and **All-Slow** (90.0%). It also retains strong monotonicity ($\rho = 0.893$), indicating that compute scales with problem difficulty rather than being spent uniformly.

Interestingly, full CoAR attains the highest monotonicity ($\rho = 0.902$) but worse accuracy and higher token cost. In our current implementation, the added overhead comes from

Table 1. Main results on CoAR-Math. Adaptive local allocation yields the best trade-off between accuracy and computation.

Method	Pass@1	Avg Tokens	REP	Mono ρ
All-Fast	52.5%	14	–	0.892
All-Slow	90.0%	99	5.36	0.432
Adaptive-Indep	**95.0%**	100	**6.04**	0.893
CoAR	60.0%	161	0.66	**0.902**

Table 2. Accuracy and average tokens by difficulty split.

Accuracy by split				
Method	EE	EH	HE	HH
All-Fast	100%	40%	40%	30%
All-Slow	100%	100%	80%	80%
Adaptive-Indep	**100%**	**100%**	**90%**	**90%**
CoAR	100%	50%	60%	30%
Average tokens by split				
Method	EE	EH	HE	HH
All-Fast	10	14	15	19
All-Slow	79	85	113	120
Adaptive-Indep	17	91	106	186
CoAR	18	142	179	305

confidence estimation and the bookkeeping around potential propagation, while the propagation event itself rarely triggers. This makes the experiment especially informative: cooperative coordination is not automatically beneficial; it only helps if the uncertainty signal is reliable.

4.5. Difficulty-Split Analysis

Table 2 breaks performance down by difficulty split. **Adaptive-Indep** remains strong across all regimes, while **All-Slow** wastes compute on easy-easy instances where every method reaches 100% accuracy. On the token side, adaptive allocation is sharply difficulty-sensitive: it uses 17 tokens on EE but 186 tokens on HH, a $10.9\times$ increase that closely matches the change in task difficulty.

4.6. Per-Agent Behavior and Failure Mode

Table 3 reveals the main failure mode of full CoAR. If propagation is not triggered when upstream answers are wrong, the additional machinery only increases overhead and introduces extra opportunities for brittle behavior. In our experiments, the model frequently self-reports confidence above 0.6 even when it is incorrect, so downstream re-reasoning is almost never activated.

This finding has practical implications beyond language agents. In V2X perception, an upstream vehicle may be confidently wrong about an obstacle. In robot swarms, an explorer may overstate confidence in a noisy map update. In both cases, calibration is a prerequisite for uncertainty-driven coordination. Sampling-based self-consistency [15] or learned confidence predictors may be more suitable than direct self-assessment.

Table 3. Per-agent breakdown over the full benchmark.

Method	Agent 1 Acc	Agent 2 Acc	Total Tokens
All-Fast	75.0%	55.0%	568
All-Slow	90.0%	90.0%	3,971
Adaptive-Indep	**95.0%**	**95.0%**	3,995
CoAR	80.0%	62.5%	6,431

5. Discussion

The most important empirical result is not that coordination failed, but that local adaptive allocation worked extremely well. Before introducing communication rules, a cooperative system should already know how to spend more compute on harder local sub-problems. Otherwise, downstream coordination is built on a weak foundation.

Although we study a controlled compositional math benchmark, the framework maps naturally to cooperative embodied settings. Nearby vehicles in traffic form a dynamic dependency graph through shared perception; robot teams share map fragments, plans, and state estimates; multi-agent foundation-model systems exchange textual and visual intermediate outputs. In all of these settings, uncertainty-triggered compute escalation provides a principled alternative to always-on expensive reasoning.

Our benchmark contains only 40 instances and uses single-run evaluation. The study is therefore best viewed as a controlled systems analysis rather than a definitive large-scale benchmark. We also evaluate a rule-based propagation mechanism instead of learning when and how much to propagate. Finally, the current experiments are language-based; future embodied evaluations in CARLA- or Habitat-like environments would test generality under perception noise and temporal feedback.

6. Conclusion

We presented CoAR, a framework for adaptive reasoning-budget allocation in cooperative multi-agent systems. CoAR combines local difficulty-aware allocation, confidence propagation, and triggered re-reasoning, and we proposed cooperative monotonicity and cooperative compositionality as complementary properties for evaluating such systems. Experiments on a controlled two-agent benchmark show that adaptive local allocation already provides large gains in both efficiency and accuracy, while explicit coordination only helps when uncertainty signals are calibrated. We view this as the central takeaway: cooperative reasoning is not just a communication problem, but a computation-allocation problem whose success depends critically on reliable uncertainty estimation.

References

[1] Weize Chen, Yusheng Su, Jianzhu Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. *arXiv preprint arXiv:2308.10848*, 2024. 2

[2] Abhishek Das, Theo Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Michael Rabbat, and Joelle Pineau. Tarmac: Targeted multi-agent communication. In *International Conference on Machine Learning*, 2019. 2

[3] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 2

[4] Marco Dorigo, Mauro Birattari, and Guy Theraulaz. Swarm robotics: Past, present, and future. *Proceedings of the IEEE*, 109(7):1152–1165, 2021. 2

[5] Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *International Conference on Machine Learning*, 2024. 2

[6] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017. 2

[7] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuhong Cheng, Chaoyi Zhang, et al. Metagpt: Meta programming for a multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 2024. 2

[8] Yue Hu, Sihai Fang, Zimu Lei, Yiqi Zhong, Siheng Chen, et al. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. In *Advances in Neural Information Processing Systems*, 2022. 2

[9] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. 2

[10] Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for "mind" exploration of large language model society. In *Advances in Neural Information Processing Systems*, 2023. 2

[11] Niklas Muennighoff et al. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025. 1, 2

[12] Eduardo Salas, Dana E. Sims, and C. Shawn Burke. Is there a "big five" in teamwork? *Small Group Research*, 36(5): 555–599, 2005. 2

[13] Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. Learning when to communicate at scale in multiagent cooperative and competitive tasks. In *International Conference on Learning Representations*, 2019. 2

[14] Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, 2016. 2

[15] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, et al. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023. 2, 4

[16] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022. 1, 2

[17] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Li, Erkang Zhu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2024. 2

[18] Runsheng Xu, Jinlong Guo, Jinle Zhou, Xiaoqi Wang, and Jiaqi Ma. Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers. In *Conference on Robot Learning*, 2022. 2

[19] Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wenqi Shao, Bin Wang, and Jiaqi Ma. V2x cooperative perception for autonomous driving: Recent advances and challenges. *arXiv preprint arXiv:2310.03525*, 2023. 2

[20] Ziheng Xu et al. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*, 2025. 2

[21] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, 2023. 1

[22] Yichi Zhang et al. Alphaone: Reasoning models thinking slow and fast at test time. *arXiv preprint arXiv:2505.24863*, 2025. 1

[23] Yichi Zhang et al. When reasoning meets its laws. *arXiv preprint arXiv:2512.17901*, 2025. 2