

---

# Everyone Contributes! Incentivizing Strategic Cooperation in Multi-LLM Systems via Sequential Public Goods Games

---

Yunhao Liang<sup>1</sup> Yuan Qu<sup>2\*</sup> Jingyuan Yang<sup>3</sup> Shaochong Lin<sup>2</sup> Zuo-Jun Max Shen<sup>2,4</sup>

<sup>1</sup>The University of Hong Kong, Shenzhen Institute   <sup>2</sup>The University of Hong Kong

<sup>3</sup>George Mason University   <sup>4</sup>University of California, Berkeley

## Abstract

Coordinating multiple large language models (LLMs) to solve complex tasks collaboratively poses a fundamental trade-off between the computation costs and collective performance compared with individual model. We introduce a novel, game-theoretically grounded reinforcement learning (RL) framework, the Multi-Agent Cooperation Sequential Public Goods Game (MAC-SPGG), to systematically incentivize cooperation in multi-LLM ensembles. In MAC-SPGG, LLM agents move in sequence, observing predecessors' outputs and updating beliefs to condition their own contributions. By redesigning the public-goods reward, effortful contributions become the unique Subgame Perfect Nash Equilibrium (SPNE), which eliminates free-riding under traditional SPGG or PGG. Its sequential protocol replaces costly round-based information exchanges with a streamlined decision flow, cutting communication overhead while retaining strategic depth. We prove the existence and uniqueness of the SPNE under realistic parameters, and empirically show that MAC-SPGG-trained ensembles outperform single-agent baselines, chain-of-thought prompting, and other cooperative methods, even achieving comparable performance to large-scale models across reasoning, math, code generation, and NLP tasks. Our results highlight the power of structured, incentive-aligned MAC-SPGG cooperation for scalable and robust multi-agent language generation.

## 1 Introduction

Multi-LLM ensembles can outperform single models but require effective coordination. Prior paradigms span debate and strategic reasoning [Cheng et al., 2024, Du et al., 2024, He et al., 2023, Liang et al., 2024, Yi et al., 2025a] and cooperative role/vote frameworks [Li et al., 2024, 2023, Hong et al., 2024, Chen et al., 2024]. The core challenge is to retain performance gains while curbing communication cost.

We propose *MAC-SPGG*, a two-phase, game-theoretic RL framework that formalizes sequential collaboration: agents act in order, observe predecessors, and adapt contributions via incentives. Unlike coordinator-based ensembles, MAC-SPGG induces cooperation through a public-goods reward, yielding stable equilibria with reduced overhead; see Figure B.1.

In our framework, we prove that under reasonable conditions in the inference phase (the SPGG part), a unique *Subgame Perfect Nash Equilibrium (SPNE)* emerges, shifting agents' behaviors from free-

---

\*Correspondence: yuanqu@hku.hk, jyang53@gmu.edu

riding in traditional public goods games toward positively cooperative participation. By embedding such theoretically guaranteed equilibrium strategies—largely absent in existing debate-, voting-, or heuristic-based coordination methods [Du et al., 2024, Li et al., 2024, Chen et al., 2023a,b]—our framework achieves both strategic depth and significantly reduced communication overhead compared to iterative information exchanges.

In the optimization phase (the learning part), our training process empirically demonstrates the effectiveness of MAC-SPGG: multi-LLM ensembles are robustly directed toward cooperative equilibria, consistently outperforming single-agent baselines, Chain-of-Thought prompting [Wei et al., 2022], and other cooperative frameworks across four diverse tasks, including code generation (HumanEval), factual knowledge (MMLU), mathematical reasoning (GSM8K), and natural language understanding (SummEval). We further assess two Bayesian belief update strategies, *Partial Observation* (PO) and *Full Observation* (FO), reflecting varying levels of inter-agent transparency, and find that optimal information sharing is context-dependent, with minimal transparency sometimes yielding superior outcomes. Overall, our contributions are twofold: (i) we propose a theoretically grounded MAC-SPGG framework for structured multi-LLM cooperation, whose equilibrium-driven behaviors are guaranteed by the existence and uniqueness of the SPNE; and (ii) we provide comprehensive empirical validation and ablation studies showing its consistent superiority over single-agent and cooperative baselines.

## 2 Related Work

Our work connects multi-agent LLM collaboration and mechanism design. Prior collaboration paradigms include role-playing [Li et al., 2023, Hong et al., 2024], voting/consensus [Wang et al., 2023, Park et al., 2025, Li et al., 2024], and debate-style prompting [Du et al., 2024, Liang et al., 2024, Chen et al., 2024], which improve robustness but typically lack guarantees. On the mechanism-design side, studies probe LLM rationality and embed structured incentives [Mao et al., 2025, Hua et al., 2024, Cheng et al., 2024, Sel et al., 2024], yet convergence and stability are seldom proven. MAC-SPGG offers a sequential, incentive-aligned alternative with SPNE guarantees, complementing these empirical frameworks.

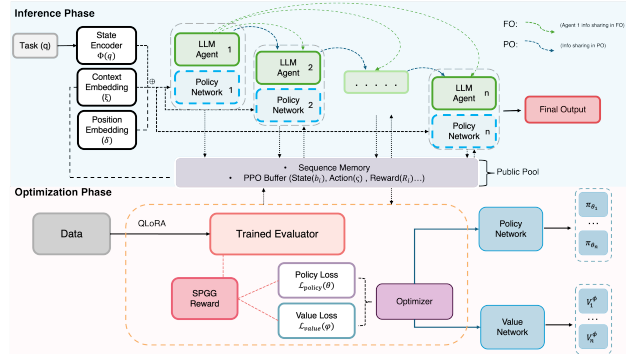


Figure 1: MAC-SPGG Framework. Top: *The Inference Phase*, where LLM agents act in sequence, conditioned on (Partial/Full) observation regimes. Bottom: *The Optimization Phase*, where SPGG rewards drive PPO updates for policy and value networks.

## 3 Method

We introduce MAC-SPGG, a multi-agent cooperation framework for sequential public goods games. The framework consists of  $n$  LLM agents performing a shared task  $q$  in a sequential, decentralized setting. Each agent  $i$  provides a contribution  $\tau_i = T_i(h_i, q)$ , with  $h_i$  denoting the information set that conditions its action, so the induced best response can be written as  $c_i^*(h_i)$ . We consider two observation modes: (1) **Partial Observation (PO)**:  $h_i^{PO} = \{\tau_{i-1}\}$ , and (2) **Full Observation (FO)**:  $h_i^{FO} = \{\tau_1, \tau_2, \dots, \tau_{i-1}\}$ . Each contribution is evaluated by a task-specific score  $c_i(\tau_i, q)$  and cost  $\ell_i(\tau_i, q, T_i)$ .

In the PO schema, agent  $i$  only observes the immediate predecessor’s contribution  $\tau_{i-1}$ , following the SPGG setting, which is similar to the sense of Markov decision process. In contrast, the agents under the FO regime have full access to the complete history of prior contributions. Although the coordinator-free mechanism of MAC-SPGG saves computation resources, the FO mode would consume more tokens than the PO mode. After all agents have committed their contribution, each contribution  $\tau_i$  is evaluated by a task-specific metric (score  $c_i(\tau_i, q)$ ) and a model-related metric (cost  $\ell_i(\tau_i, q, T_i)$ ).

### 3.1 Reward Structure and Equilibrium

Our reward  $R_i$  is rooted in the classic threshold public goods game (TPGG) [Ledyard, 1994, Anwar and Georgalos, 2023a, Gallice and Monzón, 2018, Guttman et al., 2007] and directly serves as the optimization target in training: each meta-policy  $\pi_{\theta_i}(h_i)$  is learned via PPO to maximize this  $R_i$ , ensuring alignment between the theoretical incentive and the learned behavior. Our analysis relies on standard regularity assumptions on the score and cost functions (see in Appendix C), which ensure boundedness of contributions and convexity of costs. The basic form of the MAC-SPGG structure enables collaboration among LLM agents, but it is not guaranteed to be efficient. As traditional PGG-related research has revealed, the equilibrium may collapse into a situation where no one contributes, and hence the whole task fails.

**Definition 1** (Reward with Cooperation Incentive). *The reward for agent  $i \in \{1, \dots, n\}$  in the MAC-SPGG is defined as*

$$R_i = -\ell_i(\tau_i, q, T_i) + \frac{\rho}{n} C(\vec{\tau}, q) - P \mathbf{1}(C(\vec{\tau}, q) < B(q)) + \gamma_c \frac{c_i(\tau_i, q)}{B(q)} C(\vec{\tau}, q). \quad (1)$$

Here,  $B(q)$  serves as the provision point: the final score  $C(\vec{\tau}, q)$  must exceed  $B(q)$  for the shared reward to materialize. The  $\gamma_c$  weighted cooperation term extends the standard group return by linking an agent’s own contribution  $c_i$  to the realized group performance  $C(\vec{\tau}, q)$ , thus aligning individual incentives with collective success.

**Theorem 1** (Equilibrium). *Under a reasonable cooperation coefficient  $\gamma$  and failure penalty  $P$ , where*

$$\begin{aligned} \rho &> n \cdot \max_i \ell'_i(c_{\max}), \\ \gamma &> \max_{k=2, \dots, n} \frac{\ell'_k(c_{\max}) \cdot B(q) - \rho/n}{c_{\min}/B(q)}, \text{ and} \\ P &> \left( \max_i \{\ell'_i(c_{\max})\} + \gamma \frac{c_{\max}}{B(q)} + \frac{\rho}{n} \right) \cdot (c_{\max} - c_{\min}), \end{aligned}$$

*there exists a joint strategy profile  $\mathbf{c}^*(h) = (c_1^*(h_1), \dots, c_n^*(h_n))$  that constitutes a **unique** Subgame Perfect Nash Equilibrium (SPNE),*

$$\mathbf{c}_i^* \in \arg \max_{\vec{c}} \{SPNE \text{ under } R_i\},$$

*where every agent  $i \in \{1, \dots, n\}$  contributes positively,  $c_i^* > 0$ , and the overall task would succeed  $C(\vec{\tau}, q) \geq B(q)$ .*

**Theorem 2** (Comparative Statics Analysis). *Under the MAC-SPGG equilibrium, total welfare increases with cooperation incentive  $\gamma$  and public-good sharing rate  $\rho$ , but decreases with task threshold  $B$ .*

These monotonic relationships hold as long as all LLM agents contribute non-negatively in equilibrium. Detailed proofs of Theorems 1 and 2 appear in Appendix E, and numerical verification is provided in Appendix F.

To ensure efficient optimization under the same theoretical reward  $R_i$ , we adopt an *early stopping mechanism* aligned with the success criterion of the MAC-SPGG framework rather than empirical averages. Specifically, training terminates once two external conditions are jointly satisfied: (1) the final integrator’s reward exceeds a predefined threshold,  $R_n \geq R_{\text{th}}$ ; and (2) the evaluator-assessed collective score of the final output,  $C_t = E_{\text{final}}(\tau_n, q)$ , meets or surpasses a target value,  $C_t \geq C_{\text{target}}$ . These criteria ensure that early stopping reflects task-level success rather than intermediate fluctuations, consistent with Definition 1. For stability, both the final reward and the collective score must remain within a small tolerance across consecutive episodes,  $|R_t^{\text{final}} - R_{t-1}^{\text{final}}| \leq \epsilon$  and  $|C_t - C_{t-1}| \leq \epsilon$ , so that training halts only after sustained and stable cooperative improvement.

System Category	Configuration	#Params	HumanEval	MMLU	GSM8K	SummEval (Avg)
Zero-Shot COT Single-Agent	SmolLM2-1.7B-Instruct	1.7	24.4 (-49.38)	29 (-46)	45 (-50)	4.607 (-0.12)
	Llama3.1-8B-Instruct	8	59.76 (-14.02)	57 (-18)	88 (-7)	4.638 (-0.09)
	Qwen3-8B	8	64.63 (-9.15)	66 (-9)	89 (-6)	4.677 (-0.05)
Few-Shot COT Single-Agent	SmolLM2-1.7B	1.7	29.9 (-43.88)	41 (-34)	52 (-43)	–
	Llama3.1-8B	8	72.6 (-1.18)	70 (-5)	90 (-5)	–
	Qwen3-8B	8	72.0 (-1.78)	67 (-8)	92 (-3)	–
Multi-Agent Baselines	Majority Voting	17.7	–	71 (-4)	84 (-11)	–
	Multi-Agent Debate	17.7	–	66 (-9)	86 (-9)	–
	CAMEL	16	48.78 (-24.99)	42 (-33)	88 (-7)	–
	ECON	25.7	70.73 (-3.05)	64 (-11)	89 (-6)	4.590 (-0.14)
MAC-SPGG Framework (Ours)	MAC-SPGG (PO)	17.7	67.07 (-6.71)	<b>75 (-)</b>	<b>95 (-)</b>	4.449 (-0.28)
	MAC-SPGG (FO)	17.7	<b>73.78 (-)</b>	69 (-6)	93 (-2)	<b>4.728 (-)</b>

*Note.* “–” indicates not applicable, e.g., voting-based methods cannot generate coherent outputs for HumanEval or SummEval. Ordering used in both PO and FO settings: Smol → LLaMA → Qwen.

Table 1: Performance on four benchmarks with delta (in parentheses) relative to the best MAC-SPGG setup. Metrics: HumanEval in Pass@1 (%), MMLU and GSM8K in accuracy (%), and SummEval in averaged human score (0–5).

## 4 Experiment

We evaluate MAC-SPGG on HumanEval (code), MMLU (knowledge), GSM8K (math), and SummEval (summarization). Baselines include Zero-/Few-shot CoT, Majority Voting, Debate, CAMEL, and ECON. We use three agents under PO/FO settings with heterogeneous backbones (Qwen3-8B, SmolLM2-1.7B, LLaMA3.1-8B). Details and full ablations appear in Appendix G.

**Settings.** Benchmarks: HumanEval [Chen et al., 2021], MMLU [Hendrycks et al., 2021], GSM8K [Cobbe et al., 2021], SummEval [Fabbri et al., 2021]. Baselines: Zero-/Few-shot CoT [Kojima et al., 2022, Wei et al., 2022], Majority Voting [Li et al., 2024], Debate [Du et al., 2024], CAMEL [Li et al., 2023], ECON [Yi et al., 2025b]. Setup: three sequential agents with heterogeneous backbones (Qwen3-8B, SmolLM2-1.7B, LLaMA3.1-8B) under PO/FO; hyperparameters in Appendix G.7.

**Main Results.** We showed the performance of each method across four representative evaluation tasks in Table 1. The MAC-SPGG, under both PO and FO regimes, consistently outperformed most single-agent and multi-agent baselines. To provide reference points for upper-bound performance, we included GPT-3.5 Turbo [Ye et al., 2023], GPT-4-0613 [OpenAI, 2023], and Qwen2.5-72B-Instruct [Yang et al., 2025] in a zero-shot setting, without fine-tuning. We found that MAC-SPGG achieved competitive performance with significantly fewer total parameters. Details could be found in Appendix G.6. These results highlighted the effectiveness of the cooperative mechanism in MAC-SPGG: by strategically leveraging multiple smaller models and incentivizing collaboration through game-theoretic design, the framework achieved strong performance with substantially fewer parameters. For a detailed case study, we referred readers to Appendix H.

**Ablations and Ordering Effects.** Agent ordering mildly affected task performance, with the optimal sequence depending on both the task and observation regime (see Appendix Table G.1). Mechanism and heterogeneity ablations are summarized in Appendix Table G.2. Efficiency ablation is in Appendix Figure G.1.

## 5 Conclusion

This paper presented a principled framework for structured cooperation among LLM agents, grounded in Sequential Public Goods Games (SPGG). By embedding incentive-compatible mechanisms into the interaction protocol, MAC-SPGG induced conditional cooperation and sequential adaptation—capabilities that were largely absent from existing multi-agent approaches. Empirically, it improved performance across diverse tasks and regimes while reducing redundant communication. Our findings invited further exploration into mechanism design for large-scale multi-agent LLM systems, particularly in settings involving partial knowledge, bounded rationality, or open-ended objectives. We believed this work took an essential step toward scalable, mechanism-grounded, and adaptive cooperation among foundation models.



## References

- Pengyu Cheng, Tianhao Hu, Han Xu, Zhisong Zhang, Yong Dai, Lei Han, Nan Du, and Xiaolong Li. Self-playing adversarial language game enhances LLM reasoning. In *NeurIPS*, 2024.
- Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *ICML*. OpenReview.net, 2024.
- Zhitao He, Pengfei Cao, Yubo Chen, Kang Liu, Ruopeng Li, Mengshu Sun, and Jun Zhao. LEGO: A multi-agent collaborative framework with role-playing and iterative feedback for causality explanation generation. In *EMNLP (Findings)*, pages 9142–9163. Association for Computational Linguistics, 2023.
- Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *EMNLP*, pages 17889–17904. Association for Computational Linguistics, 2024.
- Xie Yi, Zhanke Zhou, Chentao Cao, Qiyu Niu, Tongliang Liu, and Bo Han. From debate to equilibrium: Belief-driven multi-agent llm reasoning via bayesian nash equilibrium. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025a. URL <https://api.semanticscholar.org/CorpusID:279260557>.
- Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. More agents is all you need. *Trans. Mach. Learn. Res.*, 2024, 2024.
- Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: communicative agents fo "mind" exploration of large language model society. In *NeurIPS*, 2023.
- Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. Metagpt: Meta programming for a multi-agent collaborative framework. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. URL <https://github.com/geekan/MetaGPT>. Published as a conference paper at ICLR 2024.
- Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. In *ACL (1)*, pages 7066–7085. Association for Computational Linguistics, 2024.
- Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Cheng Qian, Chi-Min Chan, Yujia Qin, Ya-Ting Lu, Ruobing Xie, and et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *ArXiv*, abs/2308.10848, 2023a. URL <https://api.semanticscholar.org/CorpusID:261048935>.
- Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. *ArXiv*, abs/2309.13007, 2023b. URL <https://api.semanticscholar.org/CorpusID:262217323>.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022. URL <https://api.semanticscholar.org/CorpusID:246411621>.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *ICLR*. OpenReview.net, 2023.
- Chanwoo Park, Seungju Han, Xingzhi Guo, Asuman E. Ozdaglar, Kaiqing Zhang, and Joo-Kyung Kim. Maporl: Multi-agent post-co-training for collaborative large language models with reinforcement learning. *CoRR*, abs/2502.18439, 2025.
- Shaoguang Mao, Yuzhe Cai, Yan Xia, Wenshan Wu, Xun Wang, Fengyi Wang, Qiang Guan, Tao Ge, and Furu Wei. ALYMPICS: LLM agents meet game theory. In *COLING*, pages 2845–2866. Association for Computational Linguistics, 2025.

- Wenyue Hua, Ollie Liu, Lingyao Li, Alfonso Amayuelas, Julie Chen, Lucas Jiang, Mingyu Jin, Lizhou Fan, Fei Sun, William Wang, and et al. Game-theoretic LLM: agent workflow for negotiation games. *CoRR*, abs/2411.05990, 2024.
- Bilgehan Sel, Priya Shanmugasundaram, Mohammad Kachuee, Kun Zhou, Ruoxi Jia, and Ming Jin. Skin-in-the-game: Decision making via multi-stakeholder alignment in llms. In *ACL (1)*, pages 13921–13959. Association for Computational Linguistics, 2024.
- John O. Ledyard. Public goods: A survey of experimental research. *Public Economics*, pages 111–194, 1994. URL <https://api.semanticscholar.org/CorpusID:214607050>.
- Chowdhury Mohammad Sakib Anwar and Konstantinos Georgalos. Position uncertainty in a sequential public goods game: an experiment. *Experimental Economics*, 2023a. URL <https://api.semanticscholar.org/CorpusID:260351178>.
- Andrea Gallice and Ignacio Monzón. Cooperation in social dilemmas through position uncertainty. *ERN: Non-Cooperative Games (Topic)*, 2018. URL <https://api.semanticscholar.org/CorpusID:51852661>.
- Joel M. Guttman, Leif Danziger, and Robert McClelland. Sequential contributions to public goods. 2007. URL <https://api.semanticscholar.org/CorpusID:14784313>.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and et al. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *ICLR*. OpenReview.net, 2021.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and et al. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021.
- Alexander R. Fabbri, Wojciech Kryscinski, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir R. Radev. Summeval: Re-evaluating summarization evaluation. *Trans. Assoc. Comput. Linguistics*, 9:391–409, 2021.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *NeurIPS*, 2022.
- Xie Yi, Zhanke Zhou, Chentao Cao, Qiyu Niu, Tongliang Liu, and Bo Han. From debate to equilibrium: Belief-driven multi-agent LLM reasoning via bayesian nash equilibrium. *CoRR*, abs/2506.08292, 2025b.
- Junjie Ye, Xuanning Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, and et al. A comprehensive capability analysis of GPT-3 and GPT-3.5 series models. *CoRR*, abs/2303.10420, 2023.
- OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and et al. Qwen3 technical report. *ArXiv*, abs/2505.09388, 2025. URL <https://api.semanticscholar.org/CorpusID:278602855>.
- Guido Suurmond, Otto H. Swank, and Bauke Visser. On the bad reputation of reputational concerns. *Journal of Public Economics*, 88(12):2817–2838, 2004. doi: 10.1016/j.jpubeco.2003.10.004.
- James Andreoni. Why free ride?: Strategies and learning in public goods experiments. *Journal of Public Economics*, 37:291–304, 1988. URL <https://api.semanticscholar.org/CorpusID:17935915>.
- Paul Milgrom and Chris Shannon. Monotone comparative statics. *Econometrica*, 62(1):157–180, 1994. doi: 10.2307/2951479. URL <https://www.jstor.org/stable/2951479>.

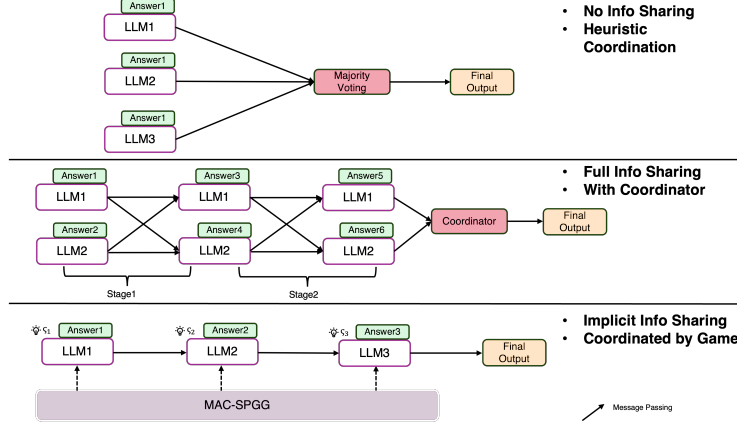


Figure B.1: Comparison of coordination mechanisms across LLM-based multi-agent systems.

Chowdhury Mohammad Sakib Anwar and Konstantinos Georgalos. Position uncertainty in a sequential public goods game: an experiment. *Experimental Economics*, 2023b. URL <https://api.semanticscholar.org/CorpusID:260351178>.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang (Eric) Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation. 2023. URL <https://api.semanticscholar.org/CorpusID:263611068>.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, and et al. Qwen2 technical report. *CoRR*, abs/2407.10671, 2024.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *ArXiv*, abs/2305.14314, 2023. URL <https://api.semanticscholar.org/CorpusID:258841328>.

## A Notation

This section summarizes the notations used throughout the paper, categorized for clarity.

## B Coordination Mechanism Illustration

This section provides a detailed comparison of coordination mechanisms across different LLM-based multi-agent systems. The figure illustrates how MAC-SPGG differs from existing approaches in terms of information flow, decision-making processes, and coordination strategies.

The comparison highlights three key aspects: (1) **Information Flow**: Traditional approaches often rely on centralized coordination or simple sequential processing, while MAC-SPGG implements a sophisticated partial observation regime that balances information sharing with computational efficiency. (2) **Decision Process**: Our framework introduces a cumulative effect mechanism where each agent’s contribution builds upon previous outputs, creating a more nuanced decision-making process. (3) **Coordination Strategy**: Unlike existing methods that may suffer from information overload or insufficient coordination, MAC-SPGG provides a structured approach to multi-agent cooperation that optimizes both performance and efficiency.

## C Assumptions and Remarks

**Assumption 1** (Score Assumption). *The score  $c_i$  of each agent  $i \in \{1, \dots, n\}$  is positive, bounded, and finite*

$$c_i \in [c_{\min}, c_{\max}], \quad \text{where} \quad 0 < c_{\min} \leq c_{\max} < \infty.$$

Symbol	Meaning	Symbol	Meaning
<b>General Notations</b>			
$n$	Total number of agents in the system	$q$	The shared task
$i, k, j$	Index for a specific agent	$\tau_i$	The textual contribution from agent $i$
$T_i$	Base Large Language Model (LLM) for agent $i$	$\vec{\tau}$	Vector of all agents' contributions
$h_i$	Observable history available to agent $i$	$h_i^{\text{PO}}$	History under Partial Observation
$h_i^{\text{FO}}$	History under Full Observation	$\mathcal{G}_i$	Generation function of agent $i$
$T_{\max}$	Maximum number of training episodes	$\epsilon$	Convergence margin
<b>Reinforcement Learning (RL) Framework</b>			
$s_t$	State vector for the RL agent at step $t$	$b_i$	Belief state of agent $i$
$\pi_{\theta_i}$	Meta-policy of agent $i$ parameterized by $\theta_i$	$V_i^\phi(b_i)$	Value function parameterized by $\phi_i$
$\vec{\zeta}_i$	Configuration vector from policy $\pi_{\theta_i}$	$A(b_i, \vec{\zeta}_i)$	Advantage function
$\mathcal{L}_{\text{PPO}}$	Clipped loss function for PPO	$R(\theta_i)$	Importance-sampling ratio in PPO
$\varepsilon$	Clipping parameter	$\lambda_{\text{value}}$	Coefficient for value loss
$\Phi(q)$	Task embedding	$\xi_i$	Contextual features (e.g., history embeddings)
$\delta_i$	Positional embedding for agent turn	$\mathcal{D}$	Experience buffer for PPO training
$\bar{R}, \bar{C}$	Avg. reward and score (early stop)	$\mathcal{H}$	Episode history log
$\theta_i^*, \phi_i^*$	Optimized policy/value params	$R_{\text{th}}, C_{\text{target}}$	Early-stopping thresholds
$r_{\text{LoRA}}, \alpha, d$	LoRA rank, scale, dropout	$\bar{R}_t, \bar{C}_t$	Reward/score at episode $t$
$\gamma_d$	Discount factor (PPO advantage)		
<b>MAC-SPGG Mathematical Model</b>			
$\chi_i$	Theoretical contribution variable (classical TPGG)	$\ell_i(\cdot)$	Cost function of agent $i$ 's contribution
$c_i$	Evaluated quality of contribution	$C(\vec{\tau}, q)$	Aggregate evaluator score
$B(q)$	Task-dependent threshold (provision point)	$B$	Threshold in classical TPGG
$R_i$	Total reward assigned to agent $i$	$S_n$	Cumulative contribution $\sum_{j=1}^n c_j$
$\gamma_c$	Cooperation coefficient	$\rho$	Multiplier for shared reward
$P$	Penalty for not reaching $B(q)$	$\mathbf{1}(\cdot)$	Indicator function (1 if true)
$U_i$	Classical threshold-PGG payoff	$R_i$	Cooperation-incentive reward
$\mathbf{c}^*$	SPNE contribution profile	$W(\cdot)$	System welfare function
$G(\cdot), f(\cdot)$	Helper functions	$\mathcal{A}^+, \mathcal{A}^-$	Success/failure regions
$t_k$	Minimum contribution avoiding penalty	$c_n^*, \tilde{c}_n$	Optimal and alternative choices
<b>Evaluator Model</b>			
$\mathcal{E}(\tau_i, q)$	Evaluator function returning $c_i$	$\mathcal{L}_{\text{eval}}$	Loss function for evaluator training
$\mathbf{r}$	Four-dimensional score vector	$r_{\text{rel}}, r_{\text{coh}}, r_{\text{flu}}, r_{\text{cons}}$	Relevance, coherence, fluency, consistency
$x_i$	Input document-summary pair	$y_t$	Target token in fine-tuning
$\mathcal{T}_{\text{score}}$	Token index set for evaluation spans		

Table A.1: Summary of Notations.

The upper bound  $c_{\max}$ , which is defined by

$$c_{\max} \equiv \sup_{\vec{\tau}} \{c_n(\tau_n(\tau_{n-1}(\cdots(\tau_1(q), q) \cdots), q), q)\},$$

can surpass the task-specific threshold  $c_{\max} \geq B(q)$ .

**Assumption 2** (Cost Assumption). *The individual cost function  $\ell_i(c_i)$  is strictly convex and twice continuously differentiable over  $[c_{\min}, c_{\max}]$ , and  $\ell'_i(c_i) > 0$ .*

**Remark 1** (No-Observation Regime). When  $h_i = \emptyset$ , agents have no cross-agent observability and act only on the task context  $q$ . This reduces to the simultaneous-move PGG setting [Suurmond et al., 2004, Andreoni, 1988]. Among the existing multi-agent LLM frameworks, ECON [Yi et al., 2025b] is the closest analogue: a central coordinator prescribes strategies to otherwise non-communicating agents. Because MAC-SPGG is coordinator-free and sequentially observable, the No-Observation regime is incompatible with the model. We omit the No-Observation setting and retain ECON as an experimental benchmark.

**Remark 2** (Cumulative Effect). Although other agents' contribution is not on the surface of  $c_n(\tau_n, q)$ , we still denote the final score  $C$  as a function of all the contributions  $\vec{\tau}$  due to the cumulative effect of the MAC-SPGG. Different from PGG, where the final performance is calculated by summing up all the contributions, the nature of multi-agent LLM tasks and prompting needs a summary step instead of concatenating the AI-generated content (AIGC) directly. In ECON [Yi et al., 2025b] or other coordinator-based frameworks, a summary agent in the last step would absorb all the others' outputs and generate the final answer. In our MAC-SPGG framework, predecessors' outputs have already been embedded into the sequential process. For instance, if we are under the FO mode, where  $c_n = T_n(h_n, q)$ ,  $h_n$  contains all the previous  $\tau_i$  information. If we are under the PO mode, we can regard the final score as

$$\begin{aligned} C(\vec{\tau}, q) &= c_n(T_n(\tau_{n-1}, q), q) \\ &= c_n(T_n(T_{n-1}(\tau_{n-2}, q), q), q) \quad \cdots \\ &= c_n(T_n(T_{n-1}(\cdots(T_1(q), q) \cdots), q), q). \end{aligned}$$

In such a context, the impact of each contribution  $\tau_i$  on the final score is not explicit, but in an iterative way.

## D Algorithm: MAC-SPGG Training

## E Proof of Theorems 1 and 2

First, we need to prove a required Lemma.

**Lemma 1** (Monotone Best Response). *Under the reward in Definition 1, the best-response contribution  $c_i^*(h_i)$*

$$c_i^*(h_i) = c_i(\tau_i^*, q)$$

*is monotonically non-decreasing in  $c_{i-1}$ ; that is,*

$$c'_{i-1} > c_{i-1} \implies c_i^*(c'_{i-1}) \geq c_i^*(c_{i-1}).$$

**Proof of Lemma 1:** We present the argument for the terminal agent  $n$ ; the same reasoning applies to any interior agent  $i$  after conditioning on the future best responses.

**Step 1:** Rewrite the payoff. Under Definition 1, agent  $n$ 's payoff is

$$\begin{aligned} R_n(c_n \mid c_{n-1}) &= -\ell_n(c_n) + \gamma \cdot \frac{c_{n-1}}{B(q)} \cdot c_n \\ &\quad + \frac{\rho}{n} \cdot (c_{n-1} + c_n) - P \cdot \mathbf{1}(c_n < B(q)). \end{aligned}$$

For convenience set

$$G(c_n, c_{n-1}) = -\ell_n(c_n) + \gamma \frac{c_{n-1}}{B(q)} c_n + \frac{\rho}{n} (c_{n-1} + c_n),$$

---

**Algorithm 1** MAC-SPGG Framework
 

---

**Require:** Initial prompt  $q$ ; base models  $\{T_i\}_{i=1}^n$ ; evaluator  $\mathcal{E}$ ; game parameters  $\rho, \gamma, P, B(q)$ ; max episodes  $T_{\max}$ ; early stopping thresholds  $R_{\text{th}}, C_{\text{target}}, \epsilon$

**Ensure:** Optimized policy and value function parameters  $\{\theta_i^*, \phi_i^*\}_{i=1}^n$

- 1: Initialize  $\{\theta_i, \phi_i\}_{i=1}^n$ , encoder  $\theta_\Phi$ , buffer  $\mathcal{D}$ , and history  $\mathcal{H}$
- 2: **for** episode  $t = 1$  to  $T_{\max}$  **do**
- 3:   Reset  $\mathcal{D} \leftarrow \emptyset, \mathcal{H} \leftarrow \emptyset$
- 4:   **for** agent  $i = 1$  to  $n$  **do** ▷ Sequential rollout
- 5:     Extract task embedding  $\Phi(q)$ , context features  $\xi_i$  and position embedding  $\delta_i$
- 6:     Construct  $b_i \leftarrow [\Phi(q); \xi_i; \delta_i]$
- 7:     Sample configuration  $\vec{\zeta}_i \sim \pi_{\theta_i}(\cdot \mid b_i)$
- 8:     Generate output  $\tau_i \leftarrow T_i(q, h_i \mid \vec{\zeta}_i)$
- 9:     Store  $(b_i, \vec{\zeta}_i, \tau_i)$  in  $\mathcal{D}$ , update  $\mathcal{H} \leftarrow \mathcal{H} \oplus \tau_i$
- 10:   **end for**
- 11:   **for** agent  $i = 1$  to  $n$  **do** ▷ Reward computation
- 12:     Evaluate quality  $c_i \leftarrow \mathcal{E}(\tau_i, q)$
- 13:     Compute reward  $R_i$ , advantage  $A_i = R_i - V^{\phi_i}(b_i)$
- 14:     Store  $(R_i, A_i)$  in  $\mathcal{D}$
- 15:   **end for**
- 16:   **Final integration:** obtain overall output  $\tau_n$  and evaluate final score

$$C_t \leftarrow \mathcal{E}_{\text{final}}(\tau_n, q), \quad R_t^{\text{final}} \leftarrow R_n$$

- 17:   **for** agent  $i = 1$  to  $n$  **do** ▷ PPO update
- 18:     Update  $\theta_i, \phi_i$  via gradient descent on  $-\mathcal{L}_{\text{PPO}}(\theta_i)$
- 19:   **end for**
- 20:   **if**  $R_t^{\text{final}} \geq R_{\text{th}}$  and  $C_t \geq C_{\text{target}}$  and
- 21:    $|R_t^{\text{final}} - R_{t-1}^{\text{final}}| \leq \epsilon$  and  $|C_t - C_{t-1}| \leq \epsilon$  **then**
- 22:     **break** ▷ Early stopping based on final integrator's score
- 23:   **end if**
- 24: **end for**
- 25: **return**  $\{\theta_i^*, \phi_i^*\}_{i=1}^n$

---

so that  $R_n = G(c_n, c_{n-1}) - P \mathbf{1}(c_n < B(q))$ .

**Step 2:** Increasing the differences of the smooth part. Because  $\ell_n$  is strictly convex, twice differentiable, and independent of  $c_{n-1}$ ,

$$\frac{\partial^2 G}{\partial c_n \partial c_{n-1}} = \frac{\gamma}{B(q)} > 0,$$

so  $G$  has *increasing differences* in  $(c_n, c_{n-1})$ .

**Step 3:** Region decomposition. Define regions

$$A^+: c_n \geq B(q), \quad A^-: c_n < B(q),$$

with corresponding payoffs

$$R_n^+(c_n, c_{n-1}) = G(c_n, c_{n-1}), \text{ and}$$

$$R_n^-(c_n, c_{n-1}) = G(c_n, c_{n-1}) - P.$$

Note that the penalty term is constant within each region and *jumps* only at the boundary  $c_n = B(q)$ .

**Step 4:** Monotonicity via a contradiction argument. Adapting the comparative-statics lemma in Milgrom and Shannon [1994], assume for contradiction that there exist  $c'_{n-1} > c_{n-1}$  with  $c_n^*(c'_{n-1}) < c_n^*(c_{n-1})$ . By examining the three possible region combinations  $(A^+, A^+)$ ,  $(A^-, A^-)$ ,  $(A^+, A^-)$  and exploiting

- the increasing-difference property of  $G$ ,
- the optimality conditions  $R_n^\bullet(c_n^*(\cdot), \cdot) \geq R_n^\bullet(\tilde{c}_n, \cdot)$  for any feasible  $\tilde{c}_n$ , and

- the fact that the penalty term is region-constant,

one arrives in each case at a strict inequality both  $\geq 0$  and  $\leq 0$ , a clear contradiction. Hence the assumed ordering reversal cannot occur, and  $c_n^*(\cdot)$  must be non-decreasing in  $c_{n-1}$ .  $\square$

With the help of Lemma 1, we can prove Theorem 1.

**Proof of Theorem 1:** We proceed by backward induction over agents  $i = n, n-1, \dots, 1$ . For any history  $h_{i-1} = (c_1, \dots, c_{i-1})$ , define  $S_{i-1} = \sum_{j=1}^{i-1} c_j$ .

**Step 1: Agent  $n$ 's Best Response**

Given  $h_{n-1}$ , Agent  $n$  maximizes:

$$R_n = -\ell_n(c_n) + \gamma \cdot \frac{c_{n-1}}{B(q)} \cdot c_n + \rho_n S_n - P \cdot \mathbf{1}(c_n < B(q)),$$

where  $S_n = S_{n-1} + c_n$ . We analyze two regions: Define:

$$\mathcal{A}^+ = \{c \in [c_{\min}, c_{\max}] \mid c \geq B(q)\}, \text{ and}$$

$$\mathcal{A}^- = \{c \in [c_{\min}, c_{\max}] \mid c < B(q)\}.$$

**Region  $A^+$ :**

$$R_n^+ = -\ell_n(c_n) + \gamma \cdot \frac{c_{n-1}}{B(q)} \cdot c_n + \rho(S_{n-1} + c_n).$$

The first-order derivative is:

$$\frac{dR_n^+}{dc_n} = -\ell'_n(c_n) + \gamma \cdot \frac{c_{n-1}}{B(q)} + \rho_n.$$

To ensure  $R_n^+$  is strictly increasing on  $[B(q), c_{\max}]$ , we require:

$$\min_{c_n \in [B(q), c_{\max}]} \frac{dR_n^+}{dc_n} > 0.$$

In the worst case, where  $S_{n-1} = (n-1) \cdot c_{\min}$ ,  $c_n = B(q)$ ,  $\ell'_n(c_n) = \ell'_n(c_{\max})$ :

$$\frac{dR_n^+}{dc_n} \geq -\ell'_n(c_{\max}) + \gamma \cdot \frac{c_{\min}}{B(q)} + \rho.$$

Thus, the condition is:

$$\gamma > \frac{\ell'_n(c_{\max}) - \rho}{c_{\min}/B(q)} \quad \text{if } \rho < \ell'_n(c_{\max}).$$

If  $\rho_n \geq \ell'_n(c_{\max})$ , the inequality holds trivially.

**Region  $A^-$ :**

$$R_n^- = -\ell_n(c_n) + \gamma \cdot \frac{c_{n-1}}{B(q)} \cdot c_n + \rho(S_{n-1} + c_n) - P.$$

Penalty avoidance requirement:

$$\max_{c_n \in [B(q), c_{\max}]} R_n^+ > \max_{c_n \in [c_{\min}, B(q))} R_n^-.$$

Define  $f(c_n) = -\ell_n(c_n) + \gamma \cdot \frac{c_{n-1}}{B(q)} \cdot c_n + \rho_n(S_{n-1} + c_n)$ . Then:

$$R_n^+ = f(c_n), \quad R_n^- = f(c_n) - P.$$

The critical condition is:

$$P > \max_{c_n < B(q)} f(c_n) - \max_{c_n \geq B(q)} f(c_n).$$

By the Lagrange mean value theorem:

$$|\max f - \min f| \leq [\max |f'(c_n)|] \cdot (c_{\max} - c_{\min}),$$

where

$$|f'(c_n)| \leq \ell'_n(c_{\max}) + \gamma \cdot \frac{c_{\max}}{B(q)} + \rho.$$

Thus, a sufficient condition is:

$$P > \left( \ell'_n(c_{\max}) + \gamma \cdot \frac{c_{\max}}{B(q)} + \rho \right) \cdot (c_{\max} - c_{\min}).$$

### Step 2: Agent $k < n$ 's Best Response

Assume successors play equilibrium strategies. Agent  $k$  maximizes  $R_k$  given  $h_{k-1}$ .

Region  $A^+$  ( $c_k \geq t_k$ ):

$$R_k^+ = -\ell_k(c_k) + \gamma \cdot \frac{c_{k-1}}{B(q)} \cdot c_k + \rho \cdot (S_k + (n - k) \cdot c_{\max}),$$

where  $S_k = S_{k-1} + c_k$ . The derivative is:

$$\frac{dR_k^+}{dc_k} = -\ell'_k(c_k) + \gamma \cdot \frac{c_{k-1}}{B(q)} + \rho.$$

Worst-case monotonicity, where  $S_{k-1} = (k - 1)c_{\min}$ ,  $c_k = c_{\min}$ , and  $\ell'_k(c_k) = \ell'_k \cdot (c_{\max})$ :

$$\frac{dR_k^+}{dc_k} \geq -\ell'_k(C_{\max}) + \gamma \cdot \frac{C_{\min}}{B(q)} + \rho.$$

The condition is:

$$\gamma > \frac{\ell'_k(c_{\max}) - \rho_n}{c_{\min}/B(q)}.$$

Region  $A^-$  ( $c_k < t_k$ ):

$$R_k^- = R_k^+ - P.$$

Penalty avoidance:

$$P > \max_{c_k < t_k} R_k^+ - \max_{c_k \geq t_k} R_k^+.$$

Using the mean value theorem:

$$P > \left( \ell'_k(c_{\max}) + \gamma \cdot \frac{c_{\max}}{B(q)} + \rho_n \right) \cdot (c_{\max} - c_{\min}).$$

### Step 3: Unified Parameter Conditions

For all  $k \in \{1, \dots, n\}$ , the following must hold:

1. Monotonicity:

$$\gamma > \max_{k=1, \dots, n} \frac{\ell'_k(c_{\max}) - \rho_n}{c_{\min}/B(q)}.$$

2. Penalty:

$$P > \left( \max_i \ell'_i(c_{\max}) + \gamma \cdot \frac{c_{\max}}{B(q)} + \rho_n \right) \cdot (c_{\max} - c_{\min}).$$

3. Reward positivity:

$$\rho_n > n \cdot \max_i \ell'_i(C_{\max}) \Rightarrow \ell'_k(C_{\max}) - \frac{\rho_n}{n} < 0.$$

As for the proof of uniqueness, it is still using backward induction:

**Induction Init: Agent  $n$ .**

Given history  $h_{n-1} = (c_1, \dots, c_{n-1})$ , agent  $n$  maximizes:

$$\begin{aligned} R_n(c_n) = & -\ell_n(c_n) + \gamma \cdot \frac{c_{n-1}}{B(q)} c_n \\ & + \frac{\rho}{n} (S_{n-1} + c_n) \\ & - P \cdot \mathbf{1}(c_n < B(q)). \end{aligned}$$



On  $\mathcal{A}^+$ , we compute the derivative:

$$\frac{dR_n^+}{dc_n} = -\ell'_n(c_n) + \gamma \cdot \frac{c_{n-1}}{B(q)} + \frac{\rho}{n}.$$

This is minimized at  $c_n = B(q)$  and  $c_{n-1} = c_{\min}$ :

$$\frac{dR_n^+}{dc_n} \geq -\ell'_n(C_{\max}) + \gamma \cdot \frac{c_{\min}}{B(q)} + \frac{\rho}{n} > 0.$$

Hence  $R_n$  is strictly increasing on  $\mathcal{A}^+$ , and  $\arg \max R_n^+ = \{c_{\max}\}$ .

To eliminate  $\mathcal{A}^-$ , define  $f(c) := R_n^+(c)$ . Then by the mean value theorem:

$$\max f - \min f \leq \max |f'(c)| \cdot (c_{\max} - c_{\min}),$$

and

$$|f'(c)| \leq \ell'_n(C_{\max}) + \gamma \cdot \frac{c_{\max}}{B(q)} + \frac{\rho}{n}.$$

So,

$$\max_{c \in \mathcal{A}^-} R_n(c) < \min_{c \in \mathcal{A}^+} R_n(c),$$

if  $P$  satisfies the given bound. Thus,

$$c_n^* = c_{\max}.$$

**Inductive Step: Agent  $k < n$ .**

Assume  $c_{k+1}^* = \dots = c_n^* = c_{\max}$ . Then:

$$S_n = S_{k-1} + c_k + (n - k)c_{\max}.$$

Let  $t_k$  denote the minimal contribution required by agent  $k$  to avoid penalty under history  $h_{k-1}$ , i.e.,

$$t_k = \max \{c_{\min}, B(q) - S_{k-1} - (n - k) \cdot c_{\max}\}.$$

and regions:

$$\mathcal{A}_k^+ := [t_k, c_{\max}], \quad \mathcal{A}_k^- := [c_{\min}, t_k).$$

Agent  $k$  maximizes:

$$\begin{aligned} R_k(c_k) = & -\ell_k(c_k) + \gamma \cdot \frac{c_{k-1}}{B(q)} c_k \\ & + \frac{\rho}{n} (S_{k-1} + c_k + (n - k) \cdot c_{\max}) \\ & - P \cdot \mathbf{1}(S_n < B(q)). \end{aligned}$$

On  $\mathcal{A}_k^+$ :

$$\frac{dR_k^+}{dc_k} = -\ell'_k(c_k) + \gamma \cdot \frac{c_{k-1}}{B(q)} + \frac{\rho}{n}.$$

Using  $c_{k-1} = c_{\min}$ ,  $c_k = t_k \geq c_{\min}$ :

$$\frac{dR_k^+}{dc_k} \geq -\ell'_k(c_{\max}) + \gamma \cdot \frac{C_{\min}}{B(q)} + \frac{\rho}{n} > 0.$$

Thus  $R_k^+$  is strictly increasing on  $\mathcal{A}_k^+$  and  $\arg \max R_k^+ = \{c_{\max}\}$ .

Same argument shows  $\max R_k^- < \min R_k^+$  under the given condition on  $P$ , so:

$$c_k^* = c_{\max}.$$

By induction, the unique SPNE is  $\mathbf{c}^* = (c_{\max}, \dots, c_{\max})$ . □

**Proof of Theorem 2:** We study the comparative statics of the total welfare

$$W(\gamma, \rho, B) = \sum_{i=1}^n R_i(c^*; \gamma, \rho, B), \quad R_i = -c_i^* + \frac{\rho}{n} S_n + \gamma \frac{c_{i-1}}{B} c_i^*,$$

where  $c_0 \equiv 0$  and  $S_n = \sum_{j=1}^n c_j^* \geq 0$ .

**Step 1: Envelope-theorem setup.**

For each agent  $i$  the equilibrium action  $c_i^*(\gamma, \rho, B)$  maximizes  $R_i$  subject to  $c_i \in [c_{\min}, c_{\max}]$ . Let  $\theta \in \{\gamma, \rho, B\}$ . Because  $R_i$  is continuously differentiable in both  $c_i$  and  $\theta$ , and the feasible set is parameter-independent, the (Benveniste–Scheinkman) envelope theorem gives

$$\frac{\partial W}{\partial \theta} = \sum_{i=1}^n \frac{\partial R_i}{\partial \theta} \Big|_{c=c^*}$$

**Step 2: Direct partial derivatives.**

We list the explicit derivatives for each parameter:

$$\frac{\partial R_i}{\partial \gamma} = \frac{c_{i-1}}{B} c_i^*, \quad (\text{always non-negative}),$$

$$\frac{\partial R_i}{\partial \rho} = \frac{S_n}{n}, \quad (\text{identical across } i),$$

$$\frac{\partial R_i}{\partial B} = -\gamma B^{-2} c_{i-1} c_i^*. \quad (\text{always non-positive}).$$

All signs follow from  $c_{i-1}, c_i^*, \gamma, B > 0$ .

**Step 3: Aggregate effect on welfare.**

We obtain

$$\frac{\partial W}{\partial \gamma} = \frac{1}{B} \sum_{i=1}^n c_{i-1} c_i^* > 0,$$

$$\frac{\partial W}{\partial \rho} = \sum_{i=1}^n \frac{S_n}{n} = S_n > 0,$$

$$\frac{\partial W}{\partial B} = -\frac{\gamma}{B^2} \sum_{i=1}^n c_{i-1} c_i^* < 0.$$

**Step 4: Boundary validity check.**

If for some  $i$  we have  $c_i^* = c_{\min}$  or  $c_{\max}$ , then  $c_i^*$  is locally constant in a neighborhood of  $\theta$ , hence  $\partial c_i^* / \partial \theta = 0$  and the envelope argument remains intact. Therefore, the strict sign conclusions above hold regardless of whether the equilibrium is interior or boundary.  $\square$

## F Numerical Experiment of SPNE

To concretely realize SPNE in our sequential public goods game, we implement a backward induction procedure grounded in nested optimization. The core idea is that each agent anticipates the rational responses of future agents and selects their own contribution accordingly. Specifically, Agent 3 computes its best response given prior contributions, using one-dimensional numerical optimization via `scipy.optimize.minimize_scalar`. Agent 2, in turn, optimizes its action by internally calling Agent 3's response function for every hypothetical contribution. Agent 1, at the top of the sequence, embeds both lower-level solvers to simulate downstream reactions and chooses its optimal strategy accordingly.

This recursive structure—captured by the functions `optimal_c3`, `optimal_c2`, and `optimal_c1`—embeds the logic of subgame perfection and ensures equilibrium consistency across the decision tree. The final equilibrium profile  $(c_1^*, c_2^*, c_3^*) = (0.267, 1.000, 1.000)$  confirms that contribution incentives align over time. As shown in Figure F.1, cooperation is sustained before the final stage. Figure F.2 reveals that Agent 3 obtains the highest utility, benefiting from both informational advantage and minimized coordination risk.

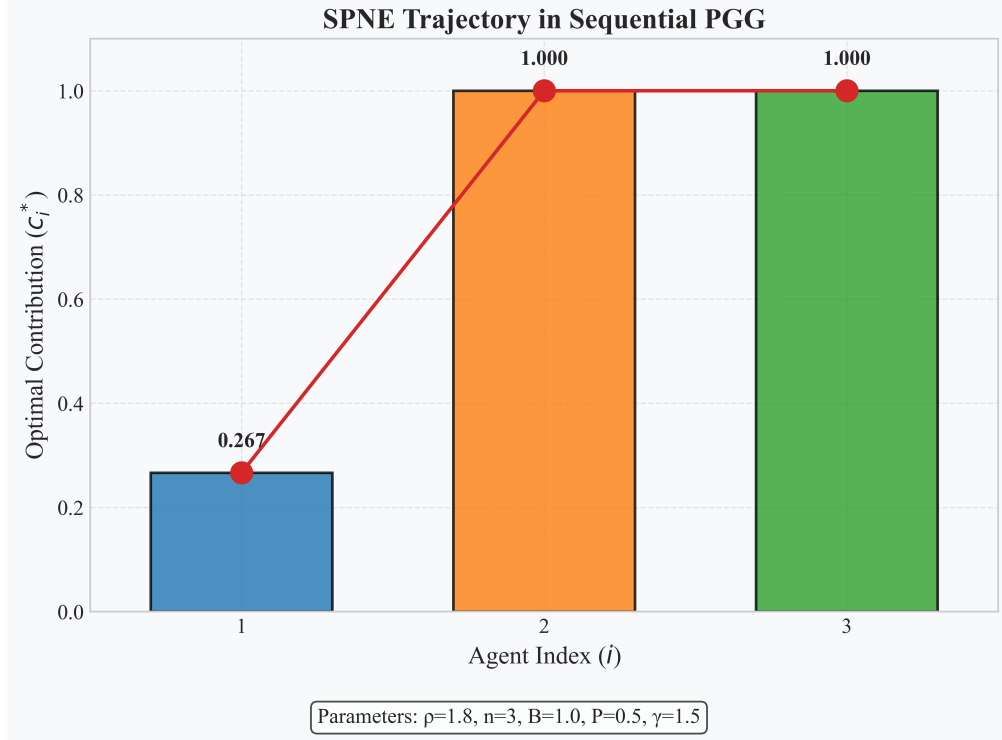


Figure F.1: SPNE contribution trajectory in sequential PGG

### F.1 Simulated Nash Trajectory Experiment

To illustrate the structure and sufficiency of the Subgame Perfect Nash Equilibrium (SPNE) under our sequential public goods game framework, we simulate a 3-agent game using backward induction. Each agent contributes sequentially based on observed history and anticipates the best responses of future agents. Based on previously established closed-form conditions, we set the parameters  $\rho = 1.8$ ,  $B = 1.0$ ,  $P = 0.5$ ,  $\gamma = 1.5$ ,  $c \in [0, 1]$ . The equilibrium strategy yields a contribution profile  $(c_1^*, c_2^*, c_3^*) = (0.267, 1.000, 1.000)$ , with total contributions exceeding the cooperation threshold.

Figure F.3 shows each agent's utility landscape, revealing strictly positive best responses at equilibrium. In Figure F.4, the cumulative contribution reaches the cooperation threshold by the second agent and is reinforced by the third, illustrating stable coordination under forward-looking reasoning.

This stylized simulation supports our theoretical claim: cooperation can emerge endogenously in MAC-SPGG, even without centralized control. We also provide a comparative statics analysis in Appendix F.

### F.2 Parameter Sampling and Analysis

We analyze three primary parameters critical to shaping the reward structure and strategic dynamics in our MAC-SPGG framework: **Cooperation coefficient**  $\gamma \in [0.5, 3.0]$ , **Reward multiplier**  $\rho \in [1.0, 3.0]$ , and **Threshold requirement**  $B \in [0.5, 2.0]$ . We sample each parameter at 25 evenly spaced points across its respective range, applying backward induction to solve for the SPNE. Equilibrium outcomes include individual utilities, total social utility, and contributions.

### F.3 Parameter and Metric Selection

We analyze three primary parameters critical to shaping the reward structure and strategic dynamics in our MAC-SPGG framework: **Cooperation coefficient**  $\gamma \in [0.5, 3.0]$ : Governs the marginal benefit of aligning contributions with preceding agents, influencing cooperative incentives. **Reward multiplier**  $\rho \in [1.0, 3.0]$ : Determines the magnitude of the total public reward pool, affecting

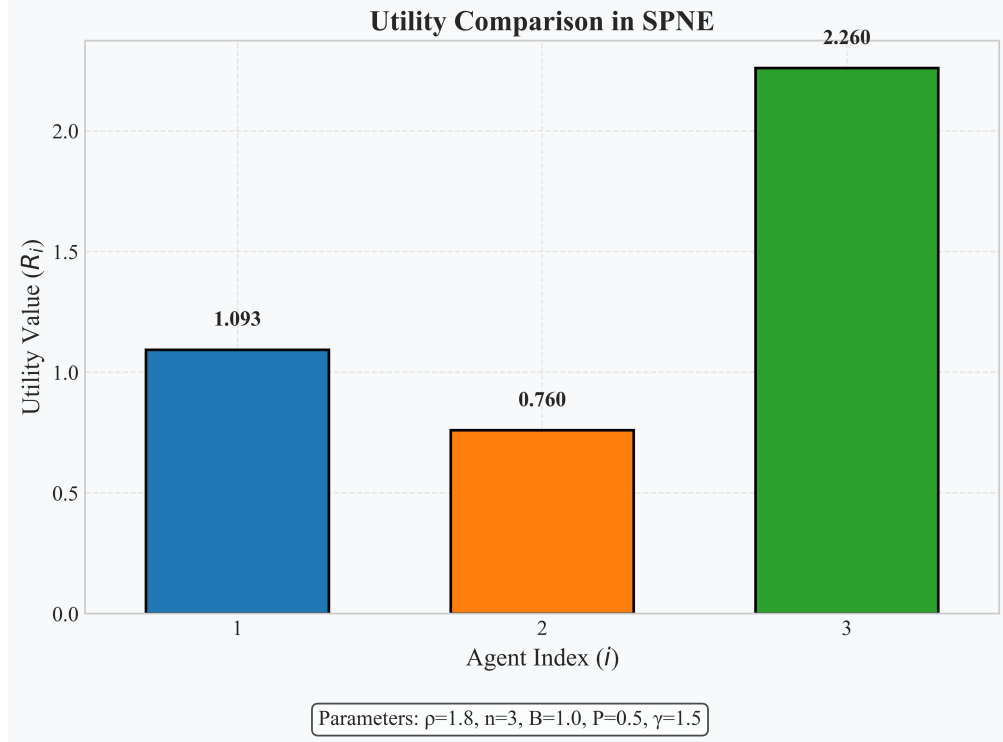


Figure F.2: Utility comparison under SPNE strategy profile

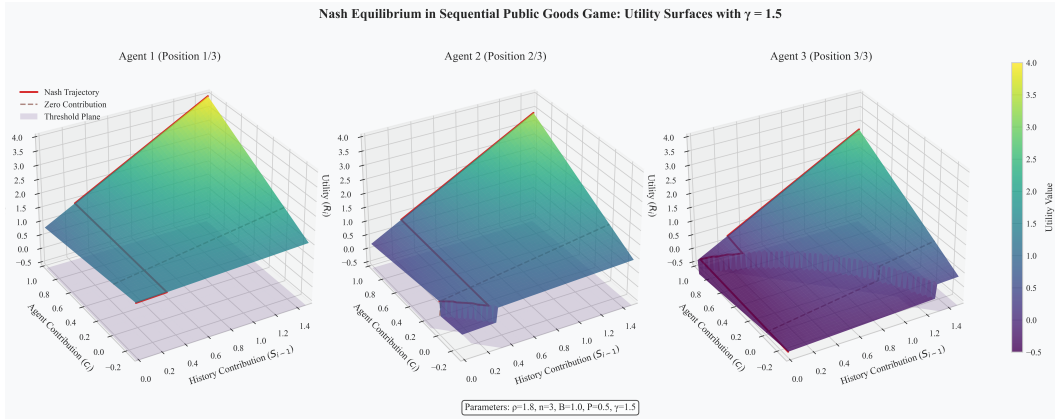


Figure F.3: Utility surfaces for Agents 1, 2, and 3 in the sequential PGG. Red curve: SPNE trajectory; shaded plane: task threshold  $B$ ; dashed line: zero-contribution baseline.

resource distribution and overall incentives. **Threshold requirement**  $B \in [0.5, 2.0]$ : Sets the minimum collective contribution necessary to realize the public good, directly impacting group coordination.

We sample each parameter at 25 evenly spaced points across its respective range while maintaining other parameters at baseline values. The penalty term  $P$  is not directly varied, as it is derived from the threshold  $B$  to maintain comparability across analyses.

After parameter selection, we apply backward induction to solve for the Subgame Perfect Nash Equilibrium (SPNE) at each sampled parameter value. The equilibrium outcomes recorded include individual utilities  $\{R_1, R_2, R_3\}$ , total social utility  $\sum_{j=1}^n R_j$ , and individual contributions  $\{c_1, c_2, c_3\}$ .

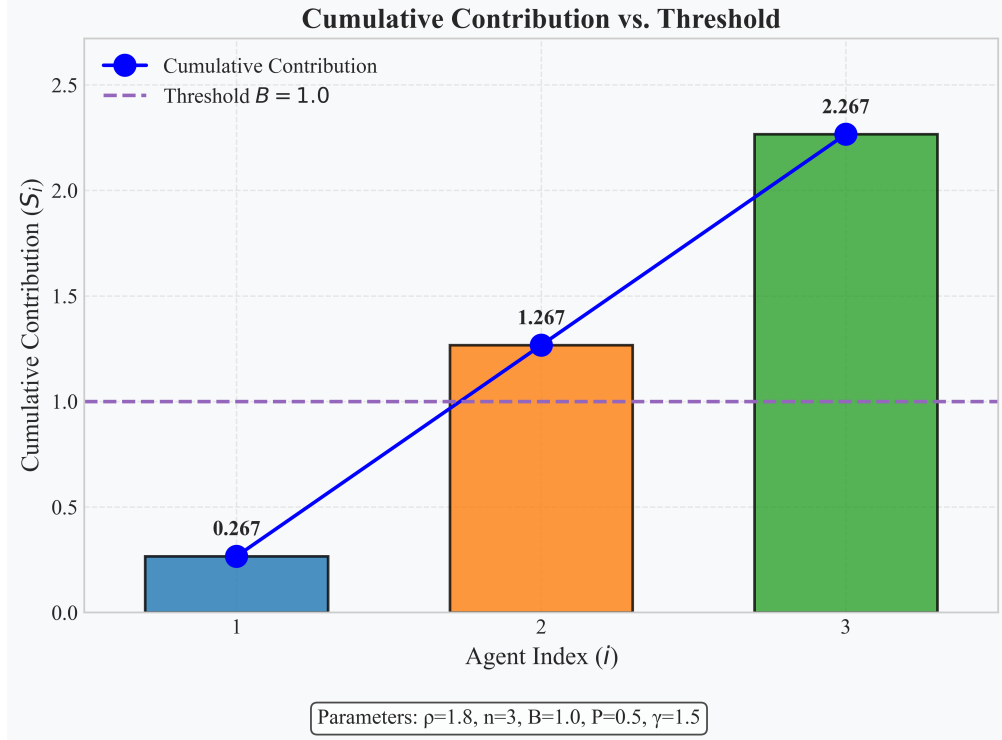


Figure F.4: Cumulative contribution trajectory. The cooperation threshold  $B = 1.0$  is reached by Agent 2.

#### F.4 Results and Observations

**Effect of Cooperation Coefficient  $\gamma$ .** As shown in Figures F.5 and F.6, both individual and total utilities exhibit strong positive correlation with  $\gamma$ . This validates our theoretical result that increasing synergy incentives amplifies cooperative behavior and leads to higher welfare. Notably, marginal utility gains taper slightly as  $\gamma$  exceeds 2.5, indicating diminishing returns in coordination incentives.

**Effect of Reward Multiplier  $\rho$ .** Figures F.7 and F.8 demonstrate a similar monotonic trend: as  $\rho$  increases, the total public good grows and agents receive higher individual rewards. However, the distribution remains sensitive to contribution ordering, and some agents benefit disproportionately depending on their sequence position and coordination exposure.

**Effect of Threshold  $B$ .** Unlike the previous parameters, increasing the task threshold  $B$  exerts a two-sided effect. As shown in Figures F.9 and F.10, agents respond by increasing their contributions to meet the higher requirement. However, this also imposes greater effort costs, leading to a net decline in total utility. This trade-off illustrates the importance of setting realistic cooperation thresholds that maintain coordination feasibility without overburdening contributors.

#### F.5 Pareto Proximity Assessment

To evaluate the allocative efficiency of our equilibrium outcome, we conduct a Monte Carlo-based test of Pareto optimality under representative parameters ( $\gamma = 1.5$ ,  $\rho = 1.8$ ,  $B = 1.0$ ), using the backward induction method described in Section F.1. We uniformly sample 10,000 alternative contribution profiles from the strategy space  $[0, 1]^3$  and compute their corresponding utility vectors under the same reward structure.

We define a profile as Pareto dominating the SPNE solution  $c^*$  if it yields weakly higher utility for all agents and strictly higher utility for at least one. Among the sampled profiles, no such dominated profile was identified. As shown in Figure F.11, this result provides numerical evidence that the SPNE outcome is not only strategically stable but also Pareto efficient within the explored strategy space.

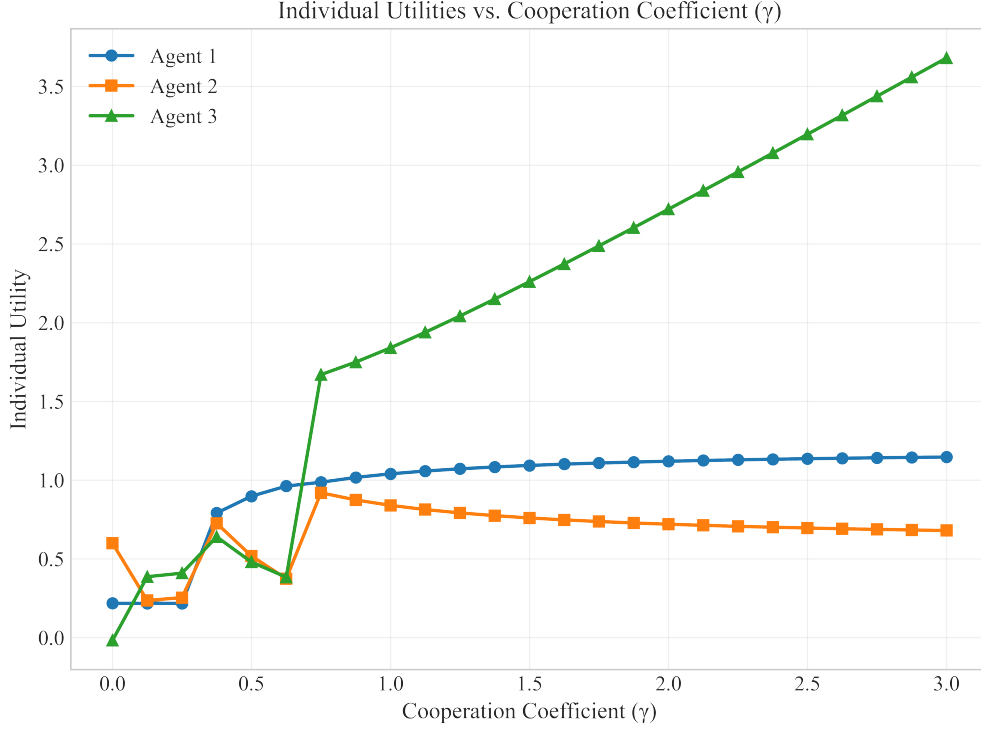


Figure F.5: Individual utilities under varying cooperation coefficient  $\gamma$ .

## G Technical Details of Section 4

This section provides additional technical details and extended formulations for the MAC-SPGG framework. The main formulation and theoretical results are presented in Section 3.

### G.1 Extended Formulation Details

For the history  $h_i$ , we have two modes of observations under the MAC-SPGG framework: (1) **Partial Observation (PO)**: The agent  $i$  can observe only the contribution from the immediately preceding agent (if any),  $h_i^{PO} = \{\tau_{i-1}\}$ , and (2) **Full Observation (FO)**: The agent  $i$  can observe all contributions made by previous agents,  $h_i^{FO} = \{\tau_1, \tau_2, \dots, \tau_{i-1}\}$ .

In the PO schema, agent  $i$  only observes the immediate predecessor’s contribution  $\tau_{i-1}$ , following the SPGG [Anwar and Georgalos, 2023b, Gallice and Monzón, 2018] setting, which is similar to the sense of Markov decision process. In contrast, the agents under the FO regime have full access to the complete history of prior contributions. Both types of observation settings exist in multi-agent LLM studies [Du et al., 2024, Wu et al., 2023]. Such a difference in information availability and resource usage will lead to distinct comprehensibility in various types of tasks in our experiment.

The score indicates the performance of the contribution, which is evaluated by a given task-specific function  $\mathcal{E}$ ,  $c_i = \mathcal{E}(\tau_i, q)$ . For instance, in multiple-choice tasks, the score represents the accuracy of the test; in more complex tasks, such as a generation task, the score is evaluated by a fine-tuned evaluator; see training details in Appendix G.7. We denote the score by  $c_i(\tau_i, q)$  to show its relevance to  $\tau_i$  and  $q$ . For the cost part, under the usage of LLM, the number of consumed tokens would be a straightforward measure of cost, and different base models  $T$  will lead to various levels of token usage.

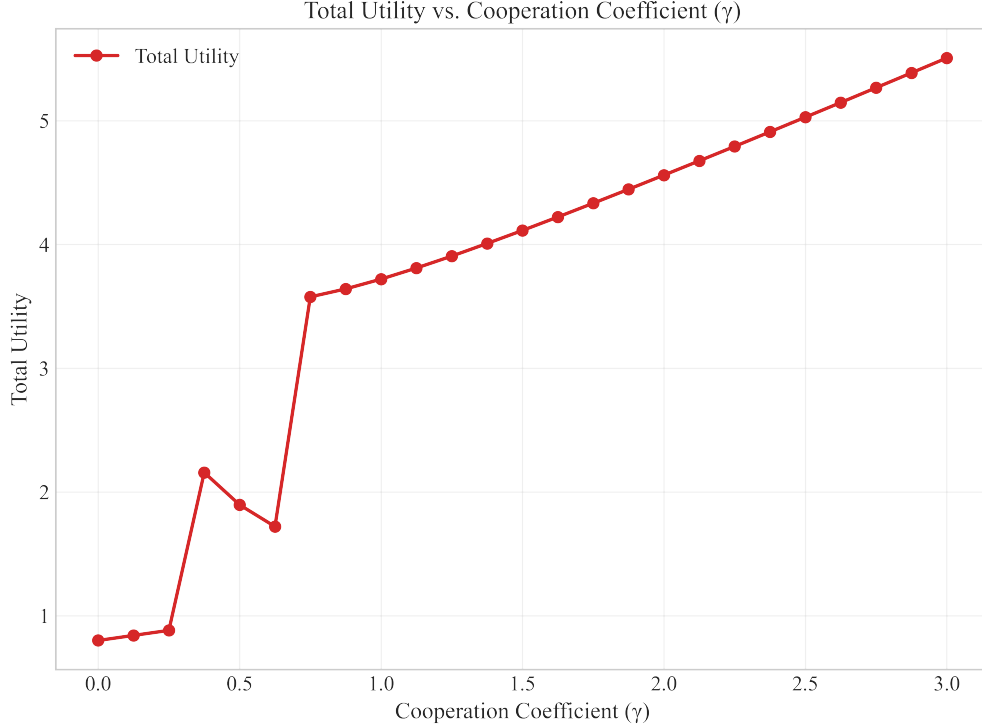


Figure F.6: Total social utility under varying cooperation coefficient  $\gamma$ .

## G.2 Learning Framework and Training Details

We learn per-agent meta-policies via decentralized PPO to optimize generation parameters dynamically. The training process uses Algorithm 1 (Appendix D) with early stopping based on reward and quality thresholds.

To ensure efficient optimization and convergence, we apply an early stopping mechanism based on the empirical performance. Specifically, training is terminated once two external criteria are jointly satisfied. First, the average episodic reward across agents exceeds a predefined threshold,  $\sum_{i=1}^n R_i/n \geq R_{\text{threshold}}$ . Second, the average evaluator-assessed output quality meets or surpasses a target value,  $\bar{C} \geq C_{\text{target}}$ . Here,  $\bar{C}$  denotes the average of final task scores  $C(\vec{\tau}, q)$  across evaluation episodes. Also, we monitor convergence stability by requiring both the average reward and quality scores to remain within a small margin  $\epsilon$  across consecutive episodes,  $|\bar{R}_{t+1} - \bar{R}_t| \leq \epsilon$  and  $|\bar{C}_{t+1} - \bar{C}_t| \leq \epsilon$ , to ensure training halts only after meaningful improvements have plateaued. This early stopping strategy ensures that agents not only achieve high collaborative performance but also maintain consistent quality in generation.

Detailed training specifications and convergence criteria are provided in Appendix G.7.

## G.3 Efficiency (Token Usage)

We analyze the computational efficiency of different frameworks by measuring token usage across various tasks. This analysis is crucial for understanding the practical deployment considerations of multi-agent systems, as token consumption directly impacts both computational cost and inference speed.

Figure G.1 demonstrates that MAC-SPGG achieves superior efficiency compared to baseline approaches. The framework’s optimized coordination mechanism reduces redundant token consumption while maintaining high performance, making it particularly suitable for resource-constrained environments. The partial observation (PO) regime shows the lowest token usage, highlighting the effectiveness of our selective information sharing strategy.

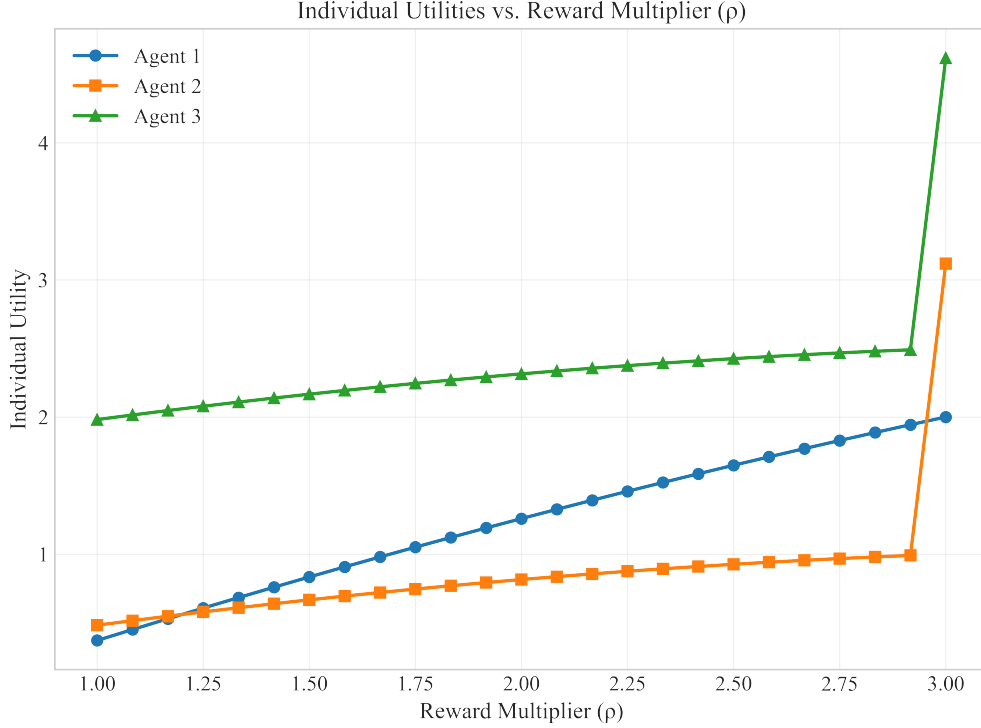


Figure F.7: Individual utilities under varying reward multiplier  $\rho$ .

Setting	Agent Order	MMLU	GSM8K
PO	Qwen → LLaMA → Smol	56	66
	Qwen → Smol → LLaMA	74	91
	Smol → Qwen → LLaMA	76	91
	LLaMA → Smol → Qwen	<b>78</b>	93
	LLaMA → Qwen → Smol	48	71
	Smol → LLaMA → Qwen	75	<b>95</b>
FO	Qwen → LLaMA → Smol	49	61
	Qwen → Smol → LLaMA	<b>77</b>	90
	Smol → Qwen → LLaMA	76	90
	LLaMA → Smol → Qwen	72	<b>96</b>
	LLaMA → Qwen → Smol	44	72
	Smol → LLaMA → Qwen	69	93

Table G.1: Agent ordering under PO and FO settings.

#### G.4 Agent Sequential Ordering Effects (Full)

This subsection presents comprehensive results from our investigation into how the sequential ordering of agents affects overall performance. The ordering of agents in a multi-agent system can significantly impact the quality of final outputs, as each agent’s contribution builds upon the previous agents’ work. Understanding these ordering effects is crucial for optimizing system design and deployment strategies.

Table G.1 reveals several key insights about agent ordering effects. First, the optimal ordering varies significantly between tasks: for MMLU, the sequence LLaMA → Smol → Qwen achieves the highest performance (78%) under PO, while for GSM8K, Smol → LLaMA → Qwen performs best (95%) under PO. Second, ending with smaller models (like Smol) often leads to performance degradation, as the final agent bears greater responsibility in cumulative decision-making. Third, the full observation (FO) setting does not consistently outperform partial observation (PO), suggesting that more information is not always beneficial and can sometimes introduce redundancy or distractions.



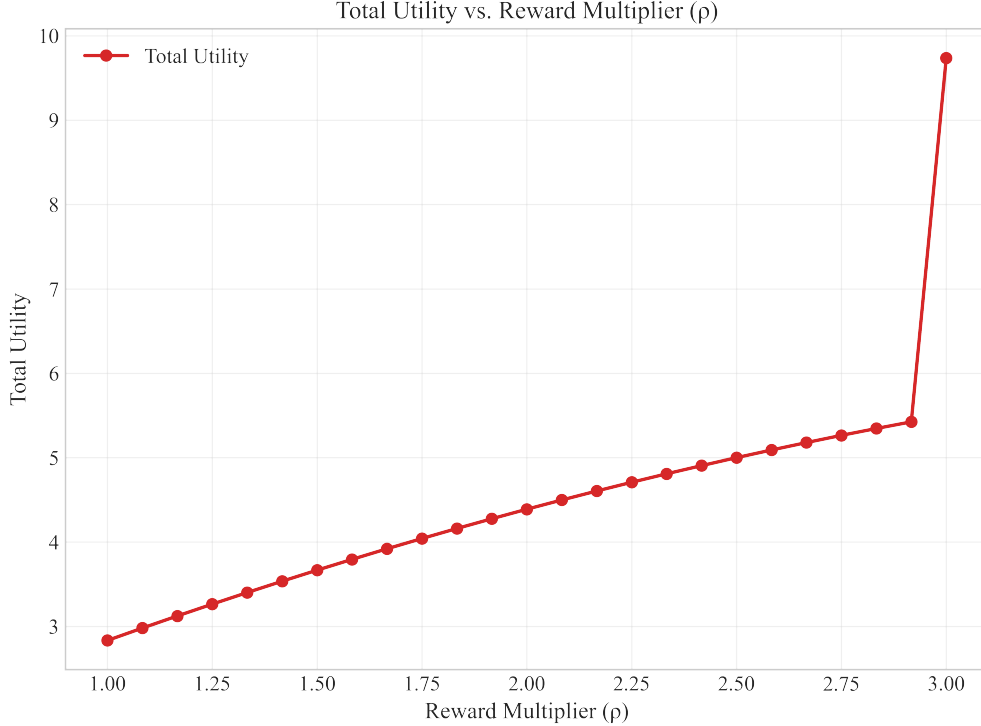


Figure F.8: Total social utility under varying reward multiplier  $\rho$ .

Obs.	Agents	Het.	SPGG	MMLU	GSM8K
PO	LLaMA + Smol + Qwen	✓	✓	<b>78</b>	93
	LLaMA + Smol + Qwen	✓		72	79
	Qwen×3		✓	<b>78</b>	<b>94</b>
	Qwen×3			71	77
FO	LLaMA + Smol + Qwen	✓	✓	72	<b>96</b>
	LLaMA + Smol + Qwen	✓		71	77
	Qwen×3		✓	<b>80</b>	95
	Qwen×3			68	74

Table G.2: Mechanism and heterogeneity ablation (complete).

## G.5 Mechanism and Heterogeneity (Full)

This subsection provides a comprehensive ablation study examining the individual contributions of the MAC-SPGG mechanism and agent heterogeneity to overall system performance. Understanding these factors is essential for designing effective multi-agent systems and determining the optimal balance between coordination mechanisms and model diversity.

Table G.2 demonstrates the critical importance of both the MAC-SPGG mechanism and agent heterogeneity. When both factors are present (heterogeneous agents with SPGG mechanism), the system achieves optimal performance across most settings. However, the MAC-SPGG mechanism alone can compensate for the lack of heterogeneity: using three identical Qwen models with SPGG achieves competitive performance (78% on MMLU, 94% on GSM8K under PO). Conversely, heterogeneous agents without the SPGG mechanism show significantly degraded performance, highlighting that coordination mechanisms are essential for leveraging model diversity effectively. This finding suggests that the MAC-SPGG framework provides a robust foundation for multi-agent cooperation that can work effectively with both homogeneous and heterogeneous agent compositions.

To facilitate fine-grained evaluation of generated summaries, we train a dedicated evaluator to assign scores on four quality dimensions—*relevance*, *coherence*, *consistency*, and *fluency*—based on a given document-summary pair [Fabbri et al., 2021]. The evaluator outputs a score vector

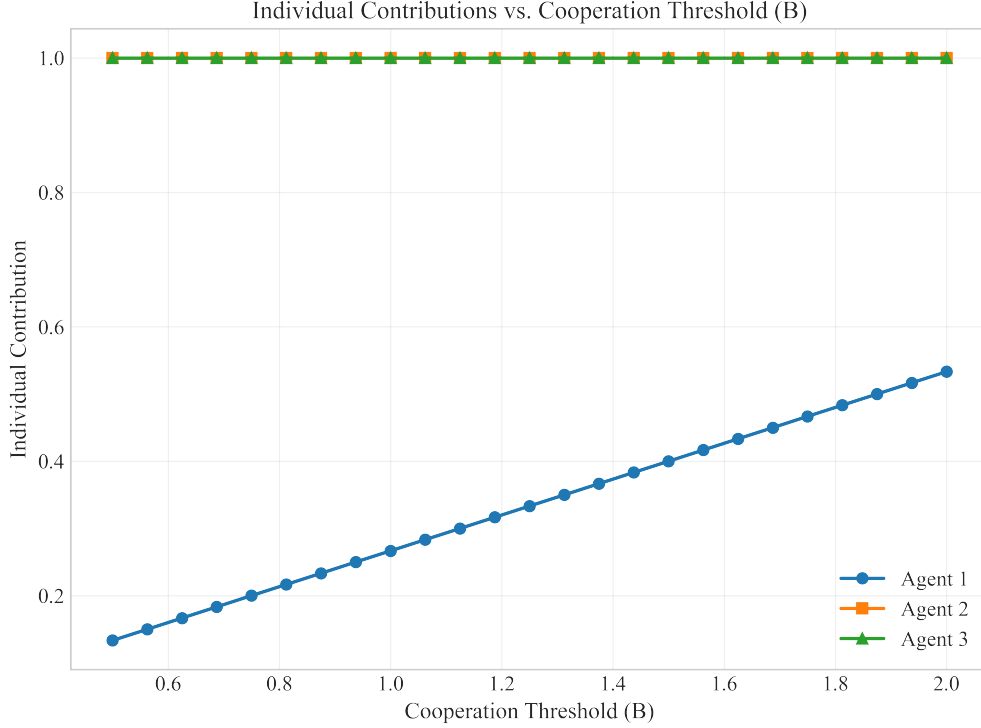


Figure F.9: Individual contributions under varying threshold  $B$ .

$\mathbf{r} = (r_{\text{relevance}}, r_{\text{coherence}}, r_{\text{consistency}}, r_{\text{fluency}}) \in [0, 5]^4$ , aligned with the scoring guidelines of the underlying dataset. These scores are used as reward signals in the reinforcement learning pipeline.

### G.5.1 Training Procedure

We frame the evaluator training task as a structured text-generation problem. Each instance in our dataset consists of a prompt comprising the source document and a candidate summary, followed by a structured output format requesting four numeric scores corresponding to the specified dimensions. During training, we only supervise numeric score tokens, masking all other tokens with the label  $-100$ , effectively constraining optimization exclusively to numeric generation.

The evaluator is a fine-tuned Qwen2.5-7B-Instruct model, quantized in 4-bit precision with Low-Rank Adaptation (LoRA). The LoRA configuration includes a rank of  $r_{\text{LoRA}} = 4$ , scaling factor  $\alpha = 8$ , and dropout rate  $d = 0.05$ , specifically targeting the model’s attention and feed-forward layers (qkv\_proj, o\_proj, gate\_up\_proj, down\_proj). The training optimizer used was AdamW with a learning rate of  $1 \times 10^{-4}$ , warmup steps set to 50, and gradient accumulation steps set to 8, resulting in an effective batch size of 16. We trained the evaluator for three epochs on the cleaned SummEval dataset [Fabbri et al., 2021], normalizing the scores to the range  $[0, 5]$ . Data was split into training and testing subsets at a 9:1 ratio with a fixed seed for reproducibility.

The training loss is computed as:

$$\mathcal{L}_{\text{eval}} = - \sum_{t \in \mathcal{T}_{\text{score}}} \log p_{\theta}^{\text{eval}}(y_t \mid x_i, y_{<t}),$$

where  $x_i$  is the input prompt (document-summary pair),  $y_t$  the target token at position  $t$ , and  $\mathcal{T}_{\text{score}}$  denotes indices corresponding specifically to numeric scores.

### G.5.2 Evaluator Performance

We evaluated the trained evaluator on the held-out SummEval test set using Mean Squared Error (MSE) and Mean Absolute Error (MAE) across the four quality dimensions. Table G.3 presents

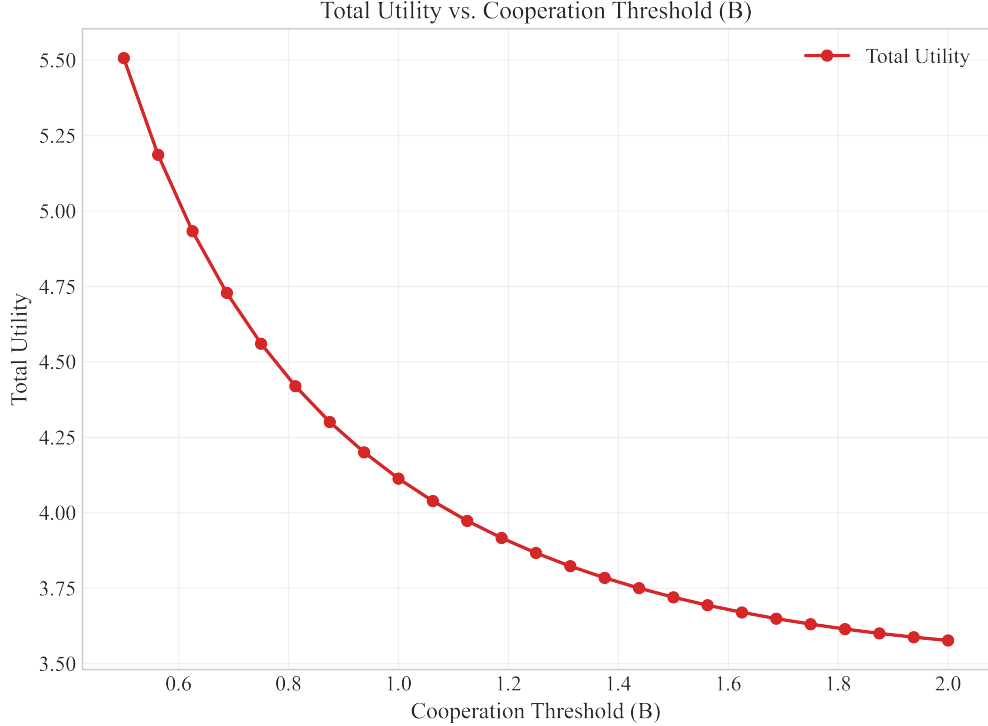


Figure F.10: Total utility under varying threshold  $B$ .

a side-by-side comparison of the pretrained and fine-tuned models. Fine-tuning led to substantial improvements, reducing overall MSE by 72.2% and MAE by 60.8%, demonstrating the effectiveness of our training strategy and the improved accuracy of the evaluator.

Metric	Pretrained Model		Fine-tuned Model	
	MSE	MAE	MSE	MAE
Relevance	1.398	0.913	0.666	0.618
Coherence	0.795	0.670	0.966	0.757
Consistency	4.096	1.737	0.539	0.227
Fluency	2.989	1.483	0.412	0.281
<b>Overall</b>	<b>2.320</b>	<b>1.201</b>	<b>0.646</b> (↓72.2%)	<b>0.471</b> (↓60.8%)

Table G.3: Evaluator performance on the SummEval test set before and after fine-tuning. Relative improvements are shown in parentheses for overall metrics.

## G.6 Comparison with Large LLMs

To further assess the efficiency of MAC-SPGG parameters, Figure G.2 compares its performance with strong proprietary models, including GPT-3.5-Turbo [Ye et al., 2023], GPT-4-0613 [OpenAI, 2023], and Qwen2.5-72B-Instruct [Yang et al., 2025]. Despite comprising only three smaller LLMs totaling 17.7B parameters, MAC-SPGG achieves performance comparable to or even exceeding these large-scale systems on certain benchmarks, notably GSM8K and SummEval.

## G.7 Technical Details of MAC-SPGG Training

For reward evaluation, we use Qwen2.5-7B-Instruct [Yang et al., 2024] as the scoring model. This evaluator is fine-tuned using QLoRA [Dettmers et al., 2023] on 4-bit quantized weights for efficient parameter adaptation.

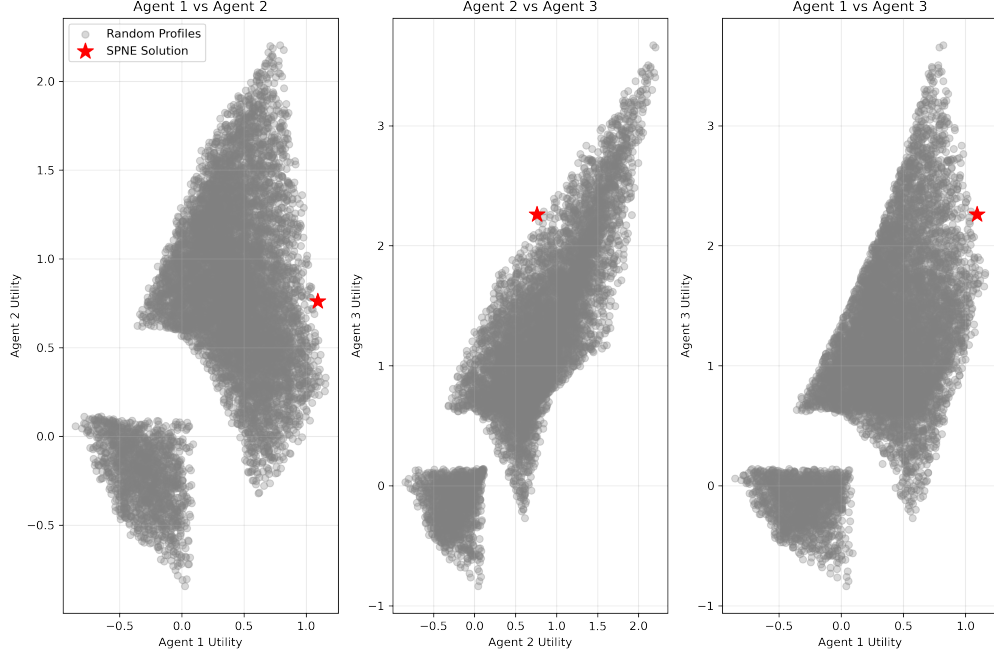


Figure F.11: SPNE utility (red star) and sampled profiles (gray) in projected utility space under ( $\gamma = 1.5$ ,  $\rho = 1.8$ ,  $B = 1.0$ ).

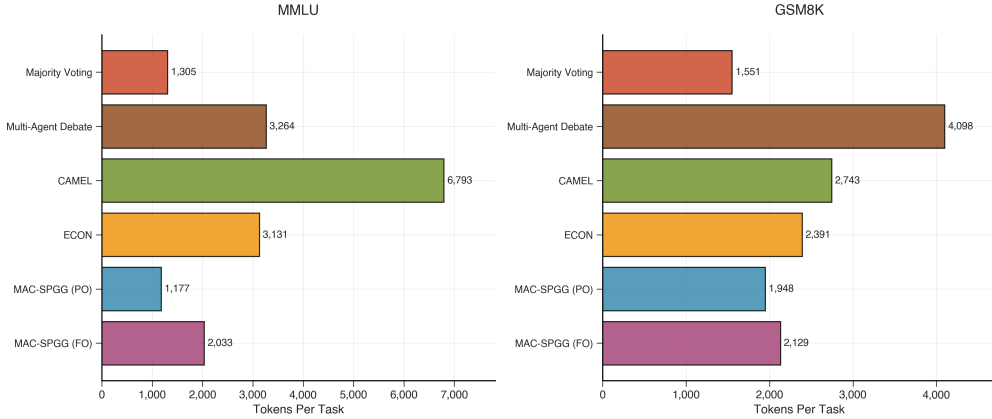


Figure G.1: Token usage per task across different frameworks. MAC-SPGG reduces tokens under both FO and PO (lowest under PO).

### G.7.1 State-to-Policy Network Architecture

To efficiently train cooperative policies in the MAC-SPGG summarization workflow, we adopt a modular and decoupled reinforcement learning architecture. A lightweight Actor-Critic policy network is trained to dynamically select optimal generation parameters for each LLM based on the evolving context of the multi-agent interaction.

Specifically, we use a pretrained bert-base-uncased model as a state encoder. For each agent at each step, we construct a comprehensive state vector  $s_t \in \mathbb{R}^{896}$  by concatenating the 768-dimensional [CLS] embedding of the source document, a 64-dimensional context vector (representing historical performance and task progress), and a 32-dimensional positional embedding indicating the agent's turn.

The policy network is a multi-layer perceptron (MLP) composed of a shared hidden layer and two task-specific heads:

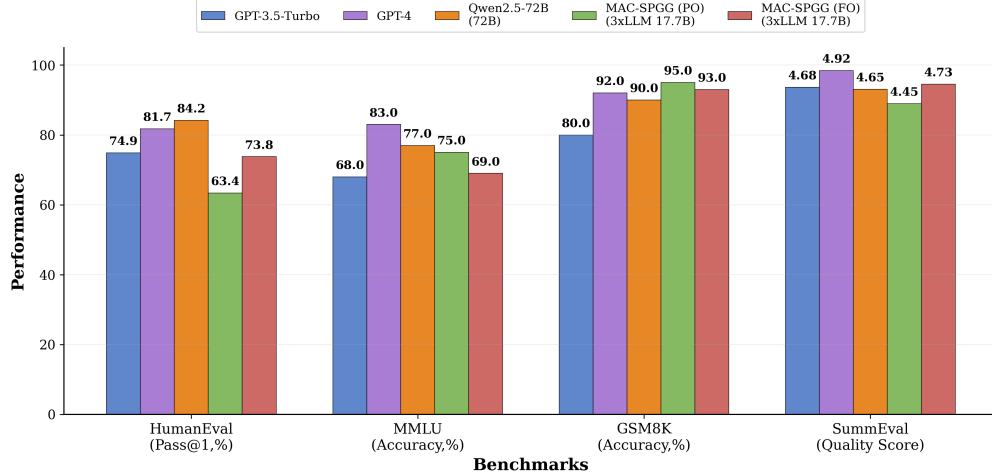


Figure G.2: Performance comparison across four benchmarks: HumanEval, MMLU, GSM8K, and SummEval. MAC-SPGG (ours) achieves competitive performance with significantly fewer total parameters.

- **Actor Head:** Predicts the mean and standard deviation for a multi-dimensional continuous action space, representing six key generation parameters: temperature, top-p, top-k, max tokens, repetition penalty, and presence penalty.
- **Critic Head:** Estimates the expected return (value) from the current state.

This architecture enables fast policy learning over the complex parameter space while avoiding the computationally prohibitive cost of backpropagation through the LLM’s forward pass.

### G.7.2 PPO Training Setup and Hyperparameters

Training was conducted on summarization tasks from the CNN/DailyMail dataset, where each document served as a MAC-SPGG-compatible episode. Each agent generated its summary using a frozen LLM guided by the parameters selected by its policy network. A trained evaluator, based on Qwen2.5-7B-Instruct, computed scalar rewards from the semantic quality of these summaries. We employed Proximal Policy Optimization (PPO) to update each agent’s actor-critic network, using the Adam optimizer with a learning rate of  $5 \times 10^{-4}$ . The PPO configuration included 4 training epochs with a mini-batch size of 16, a discount factor  $\gamma = 0.99$ , and a generalized advantage estimation (GAE) parameter  $\lambda = 0.95$ . The clip ratio was set to 0.2, the value loss coefficient to 0.5, and the entropy coefficient to 0.02, with gradient norm clipping of 0.5 and a target KL divergence of 0.015. For the MAC-SPGG reward function, we applied a task reward scaling factor  $\rho = 1.8$ , a cooperation bonus coefficient  $\gamma = 1.5$ , a success threshold  $B(q) = 0.85$ , and a failure penalty  $P = 1.5$ .

### G.7.3 Evaluator as Reward Model

We train a scalar reward model based on Qwen2.5-7B-Instruct using Low-Rank Adaptation (LoRA) on the cleaned SummEval dataset. The evaluator predicts four continuous quality dimensions — relevance, coherence, consistency, and fluency — each normalized to the  $[0, 1]$  range. These scores are averaged to produce a scalar reward for each agent’s contribution. During RL training, the evaluator remains frozen to ensure consistent and non-drifting reward signals. For evaluator training, we use a 90/10 train-test split of SummEval and constrain generation to numeric score spans via partial masking. This setup enables reward shaping with semantically meaningful, fine-grained supervision without the need for human annotators.

## G.8 Evaluation Details

**HumanEval** To assess agents’ code generation capabilities, we evaluate all models on the full HumanEval benchmark. Following standard practice, we adopt the *pass@1* metric—indicating the

percentage of problems correctly solved by the first generated solution—as our main performance indicator.

**MMLU** To evaluate MMLU, we measured the accuracy with which models were able to select the correct multiple-choice answer in each problem. We evaluated models on one hundred randomly selected MMLU questions randomly distributed across each of the subject areas.

**GSM8K** To assess mathematical reasoning ability, we evaluate models on the GSM8K dataset, which consists of grade-school math word problems requiring multi-step arithmetic and logical reasoning. We randomly selected one hundred problems from the dataset and report accuracy as the percentage of problems for which the model produces the correct final numerical answer.

**SummEval** To evaluate agents’ natural language processing ability, we use models to test all the SummEval problems and also the 1600 examples and corresponding scores given by datasets, we used them to fine-tune our evaluator.

**CNN/DailyMail** In addition to the benchmark tasks above, we trained our MAC-SPGG models on the **CNN/DailyMail** dataset (HuggingFace version 1.0.0), which contains 287,113 examples in its training split. This large-scale summarization corpus, structurally similar to SummEval, was used since SummEval’s scale is insufficient for reliable reinforcement learning.

## H Case Study

To qualitatively illustrate the collaborative dynamics fostered by our MAC-SPGG framework, we present three representative case studies in Figures H.1 and H.2. These examples involve a diverse ensemble of large language models (LLMs), including Qwen3-8B, SmolLM2-1.7B-Instruct, LLaMA3.1-8B-Instruct, and Qwen2.5-7B-Instruct. Among these, Qwen2.5-7B-Instruct is used as a *trained evaluator*, which is fine-tuned for contribution assessment tasks and kept frozen during inference (i.e., it does not generate content or update parameters). See Appendix G.7 for training details. The remaining models function as sequential contributors, collaboratively refining the output through the MAC-SPGG protocol. To ensure computational efficiency and compatibility with limited GPU memory, all models are deployed using 8-bit quantization.

These case studies highlight MAC-SPGG’s capacity to integrate diverse models into a structured collaboration framework, facilitating improvement over time even when the individual models are imperfect. This collaborative mechanism proves effective across both reasoning-intensive (MMLU) and generation-intensive (SummEval) tasks, showcasing the generality and extensibility of the proposed approach.

Question ID: high\_school\_chemistry\_121

**Formatted Task:** A 50.0 mL sample of 0.0025 M HBr is mixed with 50.0 mL of 0.0023 M KOH. What is the pH of the resulting mixture?

Options: A. 1 B. 4 C. 5 D. 11

Please analyze this multiple choice question and provide your answer as one of the options (A, B, C, or D).

**My Final Answer: A\*\*.** The pH of the resulting mixture can be calculated using the formula:  $\text{pH} = -\log_{10}[\text{H}_3\text{O}^+]$ . Since the reaction between HBr and KOH is a neutralization reaction, the resulting mixture will be neutral. In a neutral solution, the concentration of  $\text{H}_3\text{O}^+$  is equal to the concentration of  $\text{OH}^-$ . The concentration of HBr is 0.0025 M, and the concentration of KOH is 0.0023 M. To find the concentration of  $\text{H}_3\text{O}^+$ , we can use the formula:  $[\text{H}_3\text{O}^+] = (\text{Kw}/\text{Kb}) \times [\text{OH}^-]$ , where Kw is the ionization constant of water ( $1.0 \times 10^{-14}$ ) and Kb is the base ionization constant of KOH ( $1.1 \times 10^{-14}$ ). Solving for  $[\text{H}_3\text{O}^+]$ , we get  $[\text{H}_3\text{O}^+] = (1.0 \times 10^{-14}) / (1.1 \times 10^{-14}) \times (1/0.0023) = 4.74 \times 10^{-7}$  M. Now, we can calculate the pH using the formula:  $\text{pH} = -\log_{10}[\text{H}_3\text{O}^+]$ .  $\text{pH} = -\log_{10}(4.74 \times 10^{-7}) \approx 6.36$ . So, the pH of the resulting mixture is approximately 6.36.

**My Final Answer: C\*\*.** To solve this problem, we need to determine the pH of the resulting mixture after the reaction between HBr and KOH. The balanced chemical equation for this reaction is:  $\text{HBr (aq)} + \text{KOH (aq)} \rightarrow \text{KBr (aq)} + \text{H}_2\text{O (l)}$ . Since HBr is a strong acid and KOH is a strong base, the reaction will proceed completely to produce water and potassium bromide. The initial concentrations of HBr and KOH are 0.0025 M and 0.0023 M, respectively. Since the volumes of the two solutions are equal (50.0 mL each), we can assume that the total volume of the resulting mixture is 100 mL or 0.1 L. The number of moles of HBr and KOH can be calculated as follows: Moles of HBr = concentration  $\times$  volume (in L) =  $0.0025 \text{ M} \times 0.05 \text{ L} = 0.000125 \text{ mol}$  Moles of KOH = concentration  $\times$  volume (in L) =  $0.0023 \text{ M} \times 0.05 \text{ L} = 0.000115 \text{ mol}$  Since HBr is in excess, the limiting reactant is KOH. The number of moles of HBr that will react with KOH can be calculated as follows: Moles of HBr reacted = moles of KOH = 0.000115 mol. The remaining moles of HBr = initial moles of HBr - moles of HBr reacted =  $0.000125 \text{ mol} - 0.000115 \text{ mol} = 0.00001 \text{ mol}$ . The concentration of  $\text{H}^+$  ions in the resulting solution will be equal to the concentration of HBr that did not react, which is 0.00001 M. The pH of the solution can be calculated using the formula:  $\text{pH} = -\log[\text{H}^+]$ .  $\text{pH} = -\log(0.00001) = 5$ . Therefore, the pH of the resulting mixture is 5.

**My Final Answer: B\*\*.** The reaction between HBr (a strong acid) and KOH (a strong base) is a neutralization reaction. Here's the detailed, accurate analysis: 1. **Determine moles of HBr and KOH:** Moles of HBr = Molarity  $\times$  Volume =  $0.0025 \text{ M} \times 0.050 \text{ L} = 0.000125 \text{ mol}$ . Moles of KOH = Molarity  $\times$  Volume =  $0.0023 \text{ M} \times 0.050 \text{ L} = 0.000115 \text{ mol}$ . 2. **Reaction between HBr and KOH:** The balanced chemical equation is:  $\text{HBr} + \text{KOH} \rightarrow \text{KBr} + \text{H}_2\text{O}$ . This is a 1:1 molar ratio reaction. 3. **Determine excess reactant:** HBr has more moles (0.000125 mol) than KOH (0.000115 mol), so HBr is in excess. Moles of HBr remaining after reaction =  $0.000125 \text{ mol} - 0.000115 \text{ mol} = 0.000010 \text{ mol}$ . 4. **Calculate concentration of excess  $\text{H}^+$  ions:** Total volume of the mixture =  $50.0 \text{ mL} + 50.0 \text{ mL} = 100.0 \text{ mL} = 0.100 \text{ L}$ . Concentration of  $\text{H}^+$  = Moles of HBr remaining / Total volume =  $0.000010 \text{ mol} / 0.100 \text{ L} = 0.00010 \text{ M}$ . 5. **Calculate pH:**  $\text{pH} = -\log = -\log(0.00010) = -\log(1.0 \times 10^{-4}) = 4.0$ . **Correction:** The initial calculation in the original solution mistakenly concluded the answer was A ( $\text{pH} = 1$ ), but the correct pH is 4.0, which corresponds to Option B. However, the Agent\_Llama solution incorrectly calculated the pH as 5, and the Agent\_SMOLLM2 solution introduced irrelevant quadratic analysis and incorrect formulas. **Re-evaluation of the problem:** The correct pH is 4.0, which is not among the options. However, the options provided are A (1), B (4), C (5), D (11). Since 4.0 is the correct pH, the correct answer is B ( $\text{pH} = 4$ ). **Final Correction:** The original solution mistakenly concluded A ( $\text{pH} = 1$ ), but the correct answer is B ( $\text{pH} = 4$ ). The Agent\_Llama solution also incorrectly calculated the pH as 5, but the correct value is 4.0. **Final Answer: B\*\*.** (Note: The original solution mistakenly concluded A, but the correct answer is B based on the accurate calculation of  $\text{pH} = 4.0$ .)

Figure H.1: MMLU Case Study. The first agent provides an ambiguous or under-reasoned answer. Through the MAC-SPGG protocol, subsequent agents critically reassess and enhance the explanation, eventually converging on a more accurate and robust response.








	<p>Doc_id: <i>SummEval_01</i></p> <p><b>&lt;Document&gt;</b>: (CNN)A North Pacific gray whale has earned a spot in the record books after completing the longest migration of a mammal ever recorded. The whale, named Varvara, swam nearly 14,000 miles (22,500 kilometers), according to a release from Oregon State University, whose scientists helped conduct the whale-tracking study. Varvara, which is Russian for "Barbara," left her primary feeding ground off Russia's Sakhalin Island to cross the Pacific Ocean and down the West Coast of the United States to Baja, Mexico. Varvara's journey surpassed a record listed on the Guinness Worlds Records website. It said the previous record was set by a humpback whale that swam a mere 10,190-mile round trip between the "warm breeding waters near the equator and the colder food-rich waters of the Arctic and Antarctic regions." Records are nice, but Bruce Mate, the lead author of the study, thinks the long trip might say more about the whale than just its ability to swim. During her 14,000-mile journey, Varvara visited "three major breeding areas for eastern gray whales," which was a surprise to Mate, who is also the director of the Marine Mammal Institute at Oregon State University. "For her to go to Mexico," Mate said, "It's pretty strong evidence that it's where she's from." Varvara was thought to be an endangered western whale, but her ability to "navigate across open water over tremendously long distances is impressive," he said in the release, which could mean that some western gray whales are actually eastern grays. With only 150 western gray whales believed to be in existence, that number might be even lower. "Past studies have indicated genetic differentiation between the species, but this suggests we may need to take a closer look," Mate said.</p>
	<p>I'm building upon the insights provided by previous participants. My summary continues to emphasize the importance of reevaluating the classification of western and eastern gray whales based on their remarkable migrations. I propose further research into the genetic differences observed in these whales to enhance our understanding of their evolutionary history and ecological roles.</p> <p> {<i>'relevance': 2.5, 'coherence': 2.5, 'consistency': 2.5, 'fluency': 3.125</i>}</p>
	<p>Building upon the foundational contributions from Agents Qwen and SmolLM2, I would like to add that Varvara's extraordinary 14,000-mile migration not only underscores the complexity of gray whale migration patterns but also raises intriguing questions about the adaptability and resilience of these magnificent creatures in the face of climate change and habitat fragmentation. By shedding light on potential genetic connections between western and eastern gray whale populations, this discovery has significant implications for conservation efforts and the development of effective management strategies to protect these vulnerable species. Further research into the migratory behaviors and genetic diversity of gray whales will be essential to ensure their survival and continued thrive in an increasingly changing world.</p> <p> {<i>'relevance': 4.3, 'coherence': 4.7, 'consistency': 3.7, 'fluency': 4.0</i>}</p>
	<p>A North Pacific gray whale named Varvara has completed the longest mammalian migration ever recorded, traveling nearly 14,000 miles from Russia's Sakhalin Island to Baja, Mexico, surpassing previous records and challenging existing classifications of western and eastern gray whale populations. This unprecedented journey provides critical insights into potential genetic links between the two groups, raising important questions about their distinctiveness and implications for conservation strategies.</p> <p> {<i>'relevance': 5.0, 'coherence': 5.0, 'consistency': 5.0, 'fluency': 4.7</i>}</p>

Figure H.2: SummEval Case Study. A summarization task where the initial response lacks cohesion and informativeness. Subsequent agents improve sentence structure, factual completeness, and coherence. Evaluations at each stage are conducted by Qwen2.5-7B-Instruct (frozen evaluator). The final summary exhibits significantly enhanced quality as judged by the evaluator, confirming the utility of MAC-SPGG in generation tasks.