

AGENT-GSPO: COMMUNICATION-EFFICIENT MULTI-AGENT SYSTEMS VIA GROUP SEQUENCE POLICY OPTIMIZATION

Yijia Fan; Jusheng Zhang; Jing Yang; Keze Wang

Sun Yat-sen University

ABSTRACT

To combat the prohibitive communication costs of “free-for-all” multi-agent systems (MAS), we introduce **Agent-GSPO**, a framework that directly optimizes for token economy using sequence-level reinforcement learning. Agent-GSPO leverages the stable and memory-efficient Group Sequence Policy Optimization (GSPO) algorithm to train agents on a communication-aware reward that explicitly penalizes verbosity. Across seven reasoning benchmarks, Agent-GSPO not only achieves new state-of-the-art performance but does so with a fraction of the token consumption of existing methods. By fostering emergent strategies like “strategic silence,” our approach provides a practical blueprint for developing scalable and economically viable multi-agent systems.

Index Terms— Agent, Multi-Agent Systems, LLM

1. INTRODUCTION

The research frontier for Large Language Models (LLMs)[1, 2] is rapidly shifting from single-agent solvers to complex Multi-Agent Systems (MAS)[3, 2, 4]. This paradigm shift holds the promise of tackling problems far beyond the reach of any individual model. This collaborative approach has demonstrated significant potential in domains like complex reasoning and software development, where agents can pool distributed knowledge and refine solutions through iterative dialogue.

However, this collaborative promise is frequently undermined by a critical bottleneck: communication inefficiency. Many advancements[5, 6] rely on “free-for-all” communication protocols, permitting agents to broadcast information with little to no cost. This architectural choice often leads to an exponential increase in token consumption and a low signal-to-noise ratio, inundating agents with verbose, low-value exchanges. Existing methods[7, 8, 9], despite their performance gains, often overlook this severe resource cost, resulting in prohibitive operational overhead that hinders practical, large-scale deployment. We argue that the core issue is not a lack of communication, but the absence of a principled mechanism to enforce communicative discipline and resource rationality.

To address this challenge, we turn to sequence-level reinforcement learning (RL), a paradigm naturally suited for optimizing the generation of entire messages. Specifically, we leverage Group Sequence Policy Optimization (GSPO)[10], a recent advancement that stabilizes policy updates by operating on groups of entire response sequences rather than individual tokens. Unlike token-level objectives common in Proximal Policy Optimization (PPO)[11] applications, GSPO’s sequence-level clipping mechanism is inherently more stable for optimizing long, variable-length action sequences. This makes it an ideal framework for multi-agent dialogue, where an agent’s “action” is a complete utterance. Furthermore, by directly optimizing the policy on sequence-level rewards, GSPO avoids the need for an auxiliary reward or value model for every token, leading to greater memory efficiency during training.

Building on this foundation, we introduce **Agent-GSPO**, a novel framework designed to cultivate communication efficiency in LLM-based MAS. Agent-GSPO frames the generation of a message as a sequential action and directly optimizes a communication-aware reward function. This reward signal explicitly incentivizes task success while penalizing communication overhead, such as token count and conversational turns. By applying the GSPO objective, our framework encourages agents to learn a sophisticated trade-off: to “speak less but more precisely.” This process intrinsically prunes low-value chatter and fosters the emergent skill of “strategic silence,” where agents learn to withhold information that is redundant or unlikely to contribute to the team’s success.

Extensive experiments demonstrate that Agent-GSPO achieves new state-of-the-art performance across several challenging reasoning benchmarks. Crucially, this is accomplished with remarkable efficiency, consuming only a fraction of the communication tokens used by current state-of-the-art methods. Our analysis reveals that Agent-GSPO effectively learns to adapt its communication strategies, shifting from verbosity to conciseness based on the implicit value of its messages, thereby establishing a more practical and scalable framework for multi-agent collaboration.

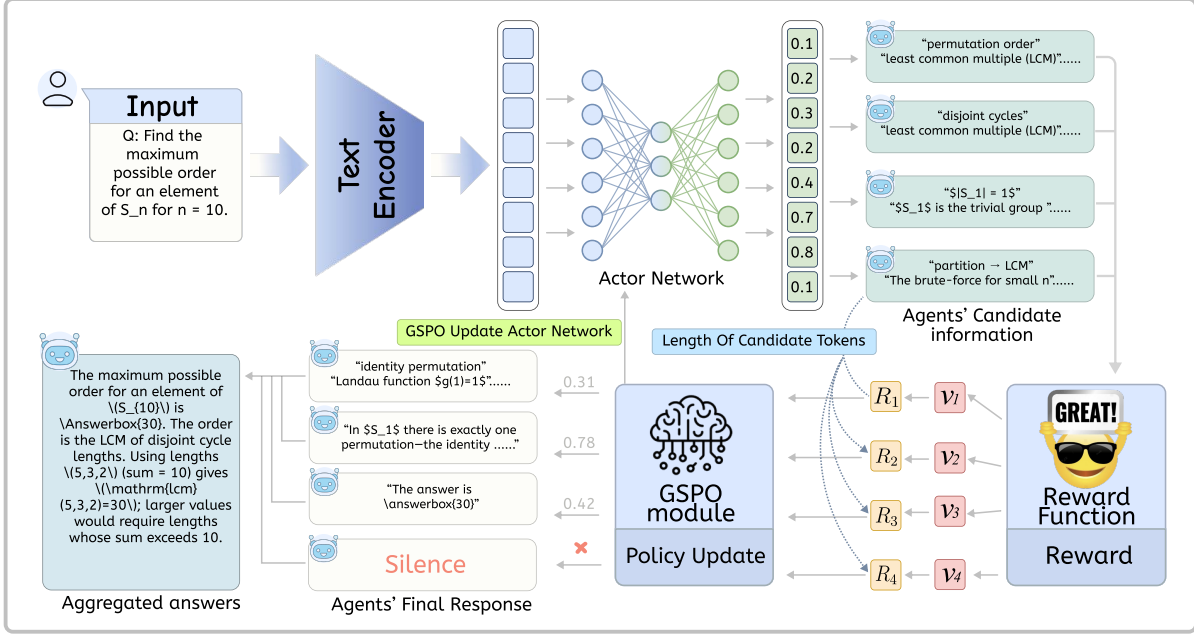


Fig. 1. Overview of the Agent-GSPO training pipeline. The Actor Network ($\pi_{\theta_{old}}$) samples a group of candidate responses. Each response receives a communication-aware reward (r), which balances task success (r_{task}) against a token-length penalty. The GSPO module processes these sequence-level rewards to update the Actor’s policy (π_{θ}), training the agent to favor concise and effective communication.

2. RELATED WORK

Communication in Multi-Agent Systems. The challenge of efficient communication in Multi-Agent Systems (MAS) has evolved significantly with the advent of LLMs. Early research focused on structured protocols like Agent Communication Languages (ACLs) and negotiation frameworks like the Contract Net Protocol to ensure clarity and manage overhead. More recently, the focus has shifted to learning emergent communication strategies. However, the “free-for-all” nature of LLM-based dialogue often leads to prohibitive token costs. To address this, current methods have explored various pruning strategies, such as gating mechanisms that decide when an agent should speak, or topological pruning that removes communication links, as seen in AgentPrune-R [12, 13]. Our work diverges from these approaches by not relying on external pruning or auction mechanisms. Instead, we formulate the problem as a direct policy optimization task, where agents intrinsically learn to be concise by optimizing a communication-aware reward signal.

Reinforcement Learning for LLMs. Reinforcement Learning (RL) has become a cornerstone for aligning LLMs with desired behaviors[14], with Proximal Policy Optimization (PPO) being the de-facto standard for both post-training alignment and tool use. However, PPO’s token-level objective can suffer from high variance and instability when optimizing long, sequential actions like natural language

messages. Furthermore, its reliance on a separate critic or value network adds significant memory and computational overhead. To overcome these limitations, recent work has introduced sequence-level policy optimization methods. Group Sequence Policy Optimization (GSPO), the algorithm we employ, stabilizes training by computing a clipped importance sampling ratio over entire sequences and uses group-wise reward normalization to estimate advantages directly, obviating the need for a critic network.

3. AGENT-GSPO

3.1. Communication-Aware Reward Function

Given an input query x , the policy π_{θ} generates a response sequence $y = (y_1, \dots, y_{|y|})$. We define a composite reward function that balances task accuracy with communication efficiency:

$$r(x, y) = r_{task}(x, y) - \lambda_{tok} \cdot \text{tokens}(y) - \lambda_{turn} \cdot \text{turns}(y) - \lambda_{rep} \cdot \text{repetition}(y), \quad (1)$$

where r_{task} measures task-specific performance (e.g., exact match, pass@1). The remaining terms penalize token usage, conversational turns, and content repetition. The coefficient λ_{tok} is the primary hyperparameter controlling token efficiency.

3.2. Sequence-Level GSPO Objective

To ensure optimization stability, GSPO operates at the sequence level. For each input x , we sample a group of G candidate responses $\{y_i\}_{i=1}^G$ from a frozen, older policy $\pi_{\theta_{\text{old}}}$. We then compute group-wise normalized advantages:

$$\hat{A}_i = \frac{r(x, y_i) - \mu_r}{\sigma_r + \epsilon}, \quad (2)$$

where $\mu_r = \frac{1}{G} \sum_{j=1}^G r(x, y_j)$ is the mean reward and σ_r is the standard deviation of rewards within the group, with ϵ as a small constant for numerical stability. To mitigate variance arising from sequence length disparities, the importance sampling ratio is normalized by the sequence length:

$$\begin{aligned} s_i(\theta) &= \left(\frac{\pi_{\theta}(y_i|x)}{\pi_{\theta_{\text{old}}}(y_i|x)} \right)^{\frac{1}{|y_i|}} \\ &= \exp \left(\frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \log \frac{\pi_{\theta}(y_{i,t}|x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x, y_{i,<t})} \right). \end{aligned} \quad (3)$$

The final GSPO objective, analogous to PPO, incorporates a clipping mechanism at the sequence level:

$$J_{\text{GSPO}}(\theta) = \mathbb{E}_{x, \{y_i\} \sim \pi_{\theta_{\text{old}}}} \left[\frac{1}{G} \sum_{i=1}^G \min \left(s_i(\theta) \hat{A}_i, \text{clip}(s_i(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_i \right) \right]. \quad (4)$$

This sequence-level clipping is crucial for stabilizing training; it prevents the large-variance updates that can arise from token-level probability ratios, especially for long sequences.

3.3. Dual Budget Constraint

To explicitly enforce an expected communication budget, B , we can treat λ_{tok} as a learnable dual variable, updated via projected gradient ascent:

$$\lambda_{\text{tok}} \leftarrow \max(0, \lambda_{\text{tok}} + \eta \cdot (\overline{\text{tokens}} - B)), \quad (5)$$

where $\overline{\text{tokens}}$ is the batch-average token consumption and η is a learning rate. This mechanism dynamically adjusts the penalty to steer the average communication cost towards the target budget.

3.4. Training Algorithm

The training procedure is summarized in Algorithm 1.

Algorithm 1 Agent-GSPO Training

- 1: Initialize policy π_{θ} , old policy $\pi_{\theta_{\text{old}}}$, and penalty λ_{tok} .
 - 2: **for** each training iteration **do**
 - 3: Sample a batch of queries $\{x_b\}_{b=1}^B$.
 - 4: **for** each query x_b **do**
 - 5: Sample G responses $\{y_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|x_b)$.
 - 6: Compute rewards $r(x_b, y_i)$ using Eq. 1.
 - 7: Compute group-wise normalized advantages $\hat{A}_i$.
 - 8: **end for**
 - 9: Compute sequence-level importance ratios $s_i(\theta)$ via Eq. 3.
 - 10: Construct the GSPO objective J_{GSPO} as in Eq. 3.2.
 - 11: Update policy parameters: $\theta \leftarrow \theta + \alpha \nabla_{\theta} J_{\text{GSPO}}$.
 - 12: Periodically update the old policy: $\pi_{\theta_{\text{old}}} \leftarrow \pi_{\theta}$.
 - 13: **if** using dual budget constraint **then** update λ_{tok} .
 - 14: **end for**
-

3.5. Emergent Communication Efficiency

Through this optimization, agents learn to intrinsically balance task performance and communication cost. As λ_{tok} increases (either manually or via the dual update), the policy shifts from verbose outputs towards concise summaries, keywords, or even abstains from communicating altogether. This emergent ‘‘strategic silence’’ is a key benefit of our framework, enabling state-of-the-art performance with substantially lower communication overhead.

4. EXPERIMENT

4.1. Experimental Setup

To evaluate Agent-GSPO, we compare it against a comprehensive suite of single-agent and multi-agent baselines. Single-agent methods include Vanilla direct prompting, Chain-of-Thought (CoT)[15], Complex CoT[16], and Self-Consistency (SC)[17]. Multi-agent system (MAS) methods feature PHP[18], LLM-Debate[7], DyLAN[19], and the current state-of-the-art in communication pruning, AgentPrune-R[12].

Our evaluation spans seven challenging benchmarks: **MMLU**[20] for general reasoning; **GSM8K**[21], **Multi-Arith**[22], **SVAMP**[23], **AQuA**[24], and **MATH-500**[25] for mathematical reasoning; and **HumanEval**[26] for code generation. To ensure a fair comparison, all experiments leverage `gpt-4-1106-preview` as the base model for all agents, maintaining consistency with prior work. The experimental setup requires collaboration for task success, as essential information is distributed among agents, making efficient communication critical. For specific hyperparameters, please refer to the Appendix. All experiments were conducted on NVIDIA A100 80GB GPUs.

Table 1. Overall performance comparison across seven reasoning benchmarks. We report accuracy (%) for all tasks except HumanEval, for which we report pass@1 (%). The best score in each column is in **bold**. Values in parentheses show absolute gain over the Vanilla baseline.

Method	MMLU	MultiArith	GSM8K	SVAMP	AQuA	HumanEval	MATH-500
<i>Single-Agent Methods</i>							
Vanilla	82.14	93.15	85.40	87.18	70.34	71.68	73.72
CoT	82.65 (+0.51)	94.79 (+1.64)	87.17 (+1.77)	88.32 (+1.14)	73.91 (+3.57)	75.52 (+3.84)	75.18 (+1.46)
ComplexCoT	83.78 (+1.64)	95.86 (+2.71)	87.62 (+2.22)	90.17 (+2.99)	77.58 (+7.24)	74.94 (+3.26)	76.85 (+3.13)
SC	82.66 (+0.52)	96.88 (+3.73)	87.93 (+2.53)	88.69 (+1.51)	75.08 (+4.74)	77.30 (+5.62)	77.02 (+3.30)
<i>Multi-Agent Methods</i>							
PHP	83.45 (+1.31)	96.41 (+3.26)	92.45 (+7.05)	90.62 (+3.44)	76.25 (+5.91)	82.96 (+11.28)	79.24 (+5.52)
LLM-Debate	83.69 (+1.55)	96.27 (+3.12)	90.23 (+4.83)	90.56 (+3.38)	77.52 (+7.18)	83.79 (+12.11)	80.15 (+6.43)
DyLAN	80.16 (-1.98)	94.27 (+1.12)	88.16 (+2.76)	87.40 (+0.22)	74.16 (+3.82)	89.70 (+18.02)	81.66 (+7.94)
AgentPrune-R	83.94 (+1.80)	96.30 (+3.15)	95.83 (+10.43)	91.68 (+4.50)	78.60 (+8.26)	90.30 (+18.62)	82.81 (+9.09)
Agent-GSPO	84.10 (+1.96)	97.20 (+4.05)	96.02 (+10.62)	91.90 (+4.72)	79.80 (+9.46)	90.70 (+19.02)	83.10 (+9.38)

Table 2. Consolidated performance comparison on MMLU and GSM8K datasets. Our Agent-GSPO, highlighted in gray, demonstrates superior or competitive performance with significantly better token efficiency.

Method	MMLU		GSM8K	
	Accuracy (%)	Token Cons.	Accuracy (%)	Token Cons.
Agent-GSPO	84.1	9.20×10^5	96.0	7.20×10^6
LLM-Debate	83.7	1.50×10^6	90.2	2.20×10^7
PHP	83.4	2.60×10^6	92.5	2.60×10^7
DyLAN	80.2	1.20×10^6	88.2	1.40×10^7
Vanilla	82.1	1.50×10^5	85.4	3.50×10^6

4.2. Performance Analysis

Agent-GSPO establishes a new state-of-the-art across seven reasoning benchmarks, demonstrating superior performance and efficiency. The framework achieves top-tier accuracy with **96.02% on GSM8K** and a **90.70% pass@1 rate on HumanEval**, alongside a significant **+9.38 point** absolute gain on the challenging MATH-500 benchmark, reaching **83.10%**. Critically, Agent-GSPO’s primary contribution lies in its exceptional communication efficiency. It secures these state-of-the-art results on GSM8K while consuming only **7.2 million tokens**, a stark contrast to the 22-26 million tokens used by competing methods for lower accuracy. This efficiency is consistent across all benchmarks, such as on MMLU where it requires only **0.92 million tokens**. Results show that optimizing a communication-aware reward enables Agent-GSPO to learn cost-effective strategies that prune verbosity, yielding more accurate, scalable, and economical collaboration.

4.3. Ablation Study

To isolate the contributions of our core components, we conducted an ablation study on the MMLU benchmark, with re-

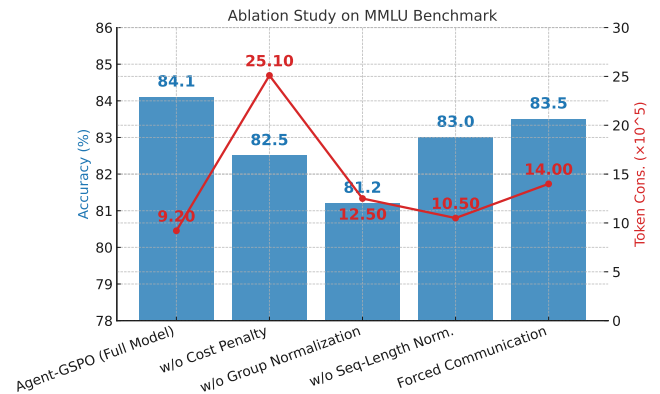


Fig. 2. Ablation study on the MMLU benchmark.

sults presented in figure 2. The analysis confirms our central hypothesis: removing the communication cost penalty ($\lambda_{\text{tok}} = 0$) causes token consumption to nearly triple while degrading accuracy, validating the need for an explicit economic disincentive. Furthermore, disabling GSPO’s group-wise advantage normalization results in the most significant accuracy drop ($\downarrow 2.9$ points), underscoring its critical role in stabilizing the high-variance, sequence-level training process.

5. CONCLUSION

In this paper, we introduced **Agent-GSPO**, a novel framework that addresses communication inefficiency in multi-agent systems by framing it as a sequence-level reinforcement learning problem. By directly optimizing a communication-aware reward using the stable and efficient Group Sequence Policy Optimization (GSPO) algorithm, our method trains agents to balance task performance with token economy. Our experiments demonstrate that Agent-GSPO establishes a new state-of-the-art across seven reasoning benchmarks with a fraction of the communication cost of prior methods. This ef-

iciency stems from the emergent skill of “strategic silence,” where agents prune verbosity, paving the way for scalable, cost-effective, and rational multi-agent systems.

6. REFERENCES

- [1] Alec Radford and Jeff Wu, “Language models are unsupervised multitask learners,” 2019.
- [2] Jusheng Zhang, Yijia Fan, Wenjun Lin, Ruiqi Chen, Haoyi Jiang, Wenhao Chai, Jian Wang, and Keze Wang, “Gam-agent: Game-theoretic and uncertainty-aware collaboration for complex visual reasoning,” 2025.
- [3] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, and Shichao Pei, “Large language model based multi-agents: A survey of progress and challenges,” 2024.
- [4] Jusheng Zhang, Kaitong Cai, Jing Yang, and Keze Wang, “Learning dynamics of vlm finetuning,” 2025.
- [5] Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O’Sullivan, and Hoang D. Nguyen, “Multi-agent collaboration mechanisms: A survey of llms,” 2025.
- [6] Jusheng Zhang, Kaitong Cai, Yijia Fan, Ningyuan Liu, and Keze Wang, “Mat-agent: Adaptive multi-agent training optimization,” 2025.
- [7] Yilun Du and Shuang Li, “Improving factuality and reasoning in language models through multiagent debate,” 2023.
- [8] Sirui Hong, Mingchen Zhuge, Jiaqi Chen, and Xiawu Zheng, “Metagpt: Meta programming for a multi-agent collaborative framework,” 2024.
- [9] Jusheng Zhang, Kaitong Cai, Yijia Fan, Jian Wang, and Keze Wang, “Cf-vm: counterfactual vision-language fine-tuning,” 2025.
- [10] Chujie Zheng, Shixuan Liu, and Mingze Li, “Group sequence policy optimization,” 2025.
- [11] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov, “Proximal policy optimization algorithms,” 2017.
- [12] Guibin Zhang and Yanwei Yue, “Cut the crap: An economical communication pipeline for LLM-based multi-agent systems,” in *The Thirteenth International Conference on Learning Representations*, 2025.
- [13] Jusheng Zhang, Kaitong Cai, Qinglin Zeng, Ningyuan Liu, Stephen Fan, Ziliang Chen, and Keze Wang, “Failure-driven workflow refinement,” 2025.
- [14] Long Ouyang, Jeff Wu, and Xu Jiang, “Training language models to follow instructions with human feedback,” 2022.
- [15] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou, “Chain-of-thought prompting elicits reasoning in large language models,” 2023.
- [16] Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot, “Complexity-based prompting for multi-step reasoning,” 2023.
- [17] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou, “Self-consistency improves chain of thought reasoning in language models,” 2023.
- [18] Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li, “Progressive-hint prompting improves reasoning in large language models,” 2024.
- [19] Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang, “A dynamic llm-powered agent network for task-oriented agent collaboration,” 2024.
- [20] Dan Hendrycks, Collin Burns, and Steven Basart, “Measuring massive multitask language understanding,” 2021.
- [21] Karl Cobbe and Vineet Kosaraju, “Training verifiers to solve math word problems,” *arXiv preprint arXiv:2110.14168*, 2021.
- [22] Subhro Roy and Dan Roth, “Solving general arithmetic word problems,” 2016.
- [23] Arkil Patel, “Are NLP models really able to solve simple math word problems?,” in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, June 2021, pp. 2080–2094, Association for Computational Linguistics.
- [24] Wang Ling and Min-Yen Kan, “Program induction by rationale generation: Learning to solve and explain algebraic word problems,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada, July 2017, pp. 158–167, Association for Computational Linguistics.
- [25] Dan Hendrycks and Collin Burns, “Measuring mathematical problem solving with the math dataset,” in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, J. Vanschoren and S. Yeung, Eds., 2021, vol. 1.
- [26] “Evaluating large language models trained on code,” 2021.