LEARNING WHO TALKS TO WHOM: ROLE-AWARE COMMUNICATION TOPOLOGY SCHEDULING FOR EFFICIENT LLM MULTI-AGENT COLLABORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent systems driven by large language models (LLMs) rely on structured communication topologies to facilitate information exchange and collaborative reasoning. However, effectively scheduling these topologies to dynamically determine *who talks to whom* and *when to communicate* remains a fundamental challenge in balancing collective reasoning performance against exponential token overhead. Existing topology scheduling approaches either enforce static communication graphs that cannot adapt to dynamic problem-solving states, or overlook agent role complementarity while failing to explicitly incorporate the synthesized graph structure into team-level decision evaluation. To tackle these limitations, we propose `ACTS`, a role-aware adaptive communication topology scheduling framework for efficient LLM multi-agent collaboration. Orchestrated within a Centralized Training with Decentralized Execution (CTDE) paradigm, `ACTS` introduces a novel dual-attention mechanism. Specifically, a *Role-conditioned Relational Attention* module dynamically selects variable-cardinality interaction partners based on agent roles and evolving execution states, while a *Topological Attention Mixer* explicitly embeds the directed communication topology into team value decomposition for precise credit assignment. Comprehensive evaluations across 9 challenging benchmarks covering code generation, multi-step reasoning, general knowledge, and competitive mathematics demonstrate that `ACTS` consistently outperforms state-of-the-art baselines, boosting average accuracy by up to 6.19 percentage points while reducing token consumption by over 15%. Code is available at `https://anonymous.4open.science/r/acts-B1F2/`.

1 INTRODUCTION

LLM-based multi-agent systems combine heterogeneous roles and complementary reasoning perspectives to solve complex tasks (Du et al., 2023; Wu et al., 2023). Structured communication supports evidence sharing and cross-agent verification but increases token consumption and inference latency (Yan et al., 2025). Exchanges that overlook agent contributions or evolving execution states may introduce redundant information (Liu et al., 2024; Wang et al., 2025). This motivates role-aware adaptive topology scheduling to determine when and between which agents communication occurs, balancing task performance against communication cost (Zhang et al., 2024; Jiang et al., 2026).

Existing communication-topology scheduling methods can be classified into static approaches, task-adaptive approaches, and multi-agent reinforcement learning (MARL)-based approaches. Static methods retain predefined interaction structures across tasks and communication rounds (Du et al., 2023; Wu et al., 2023). Task-adaptive methods generate task-specific graphs through topology induction or architecture search (Zhang et al., 2024; Sun et al., 2026), but commonly determine the complete topology before execution. Their directed links consequently remain fixed during interaction and cannot adapt to evolving local states. MARL-based topology-scheduling methods make communication decisions during execution. Agent Q-Mix(Jiang et al., 2026) is a representative MARL-based topology-scheduling method that constructs directed topologies from selected communication modes through predefined rules. Under centralized training with decentralized execution (CTDE), a QMIX (Rashid et al., 2018) monotonic mixing network combines individual utilities us-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

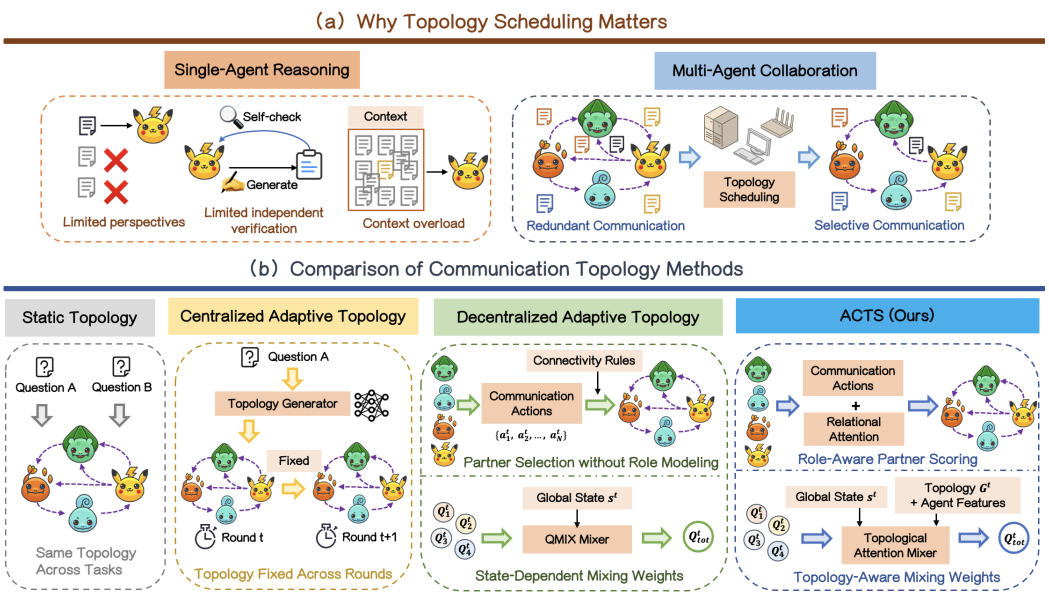

Figure 1: **Motivation and method comparison.** (a) Multi-agent collaboration addresses limitations of single-agent reasoning, while topology scheduling improves communication efficiency by regulating message exchanges. (b) Representative approaches differ in topology adaptation, partner selection, and value-function factorization. ACTS uses relational attention for role-aware partner selection and topological attention to learn topology-aware mixing weights.

ing global information during training, and agents select actions from local observations during execution. However, its partner selection is not explicitly conditioned on agent roles or evolving execution states, and its mixer does not encode the realized directed incoming and outgoing neighborhoods of individual agents. These limitations constrain adaptive partner selection and topology-specific credit assignment.

To address these limitations, this work proposes ACTS, a role-aware framework for **A**daptive **C**ommunication **T**opology **S**cheduling under centralized training with decentralized execution (CTDE). Each agent uses a local action-value function to decide *when* and *how* to communicate. Role-aware relational attention selects variable-cardinality directed partner sets from role embeddings and evolving local states, determining who communicates with whom at each round. A topological attention mixer incorporates the current directed topology into joint action-value factorization through utility weights conditioned on the global state and incoming and outgoing neighborhoods, enabling topology-aware temporal-difference (TD) credit assignment. Under a fixed pre-decision context, the monotonic mixer preserves Individual-Global-Max (IGM) consistency (Son et al., 2019), supporting decentralized greedy execution.

The main contributions are summarized as follows.

- **Role-aware communication topology scheduling.** ACTS combines communication-mode selection with role-aware relational attention to learn communication timing and variable-cardinality directed partner sets from agent roles and evolving local states. The scheduler jointly optimizes task performance and token efficiency.
- **Topology-conditioned value factorization.** A topological attention mixer learns nonnegative utility weights from the global state and directed incoming and outgoing neighborhoods for topology-aware TD credit assignment. The analysis relates graph-supported interactions to local joint-value sensitivities and establishes decentralized greedy consistency under a fixed pre-decision context.
- **Accuracy and efficiency improvements.** Evaluation covers nine benchmarks spanning four domains and two frozen backbone LLMs. Relative to Agent Q-Mix, ACTS improves macro-average accuracy by 3.18 and 6.19 percentage points and reduces average token consumption by 15.42% and 13.84% on GPT-OSS-120B and DeepSeek-V4-Flash, respectively. Ablations and communication-pattern analyses further support the benefits of role embeddings, adaptive routing, and topological mixing.

2 Related Work

2.1 Communication and Topology Control

LLM-based multi-agent systems initially relied on fixed role-based workflows, structured debates, and predefined interaction templates, including CAMEL, AgentVerse, MetaGPT, ChatDev, AutoGen and LLM-Debate (Li et al., 2023; Chen et al., 2023; Hong et al., 2023; Qian et al., 2024; Wu et al., 2023; Du et al., 2023). Subsequent methods adapt interaction structures through graph induction, multi-agent architecture search, or agent-participation control, such as G-Designer, GPTSwarm, MaAS, Assemble Your Crew, AgentDropout, GTD, and TopoDIM (Zhang et al., 2024; Zhuge et al., 2024; Zhang et al., 2025; Li et al., 2025; Wang et al., 2025; Jiang et al., 2025; Sun et al., 2026). These approaches generally do not jointly learn round-wise communication modes, directed partner selection, and topology-conditioned team-value evaluation. Agent Q-Mix introduces reinforcement-learning-based round-wise communication scheduling with a QMIX mixer (Jiang et al., 2026), but does not explicitly model agent roles or condition its value factorization on individual directed communication neighborhoods. ACTS addresses these limitations through role-aware relational attention and topological attention for directed partner selection and topology-aware value factorization.

2.2 Cooperative Multi-Agent Reinforcement Learning

Cooperative MARL provides the optimization framework for decentralized decision-making under a shared team objective. Centralized training with decentralized execution (CTDE) uses global information during training while restricting execution to local observations (Oliehoek & Amato, 2016). Value-factorization methods, including VDN, QMIX, QPLEX, and Qatten, represent the team value through individual utilities with different mixing structures (Sunehag et al., 2018; Rashid et al., 2018; Wang et al., 2021; Yang et al., 2020). Differentiable communication methods further study message passing and attention-based aggregation (Sukhbaatar et al., 2016; Foerster et al., 2016). ACTS builds on this foundation by learning discrete communication actions, role-aware interaction topologies, and topology-dependent utility weights within a CTDE framework.

3 ACTS: Dual-Attention Communication Topology Scheduling

ACTS is a CTDE framework for adaptive communication-topology scheduling in cooperative LLM agents (Oliehoek & Amato, 2016). It combines value-based mode selection with role-aware relational attention for directed partner selection, and incorporates the resulting topology into centralized value factorization through topological attention. Figure 2 illustrates the overall framework.

3.1 Problem Formulation

The proposed ACTS framework models adaptive communication topology scheduling as a cooperative multi-agent decision problem under CTDE (Rashid et al., 2018). Let $\mathcal{I} = \{1, \dots, N\}$ denote the set of cooperative LLM agents, where agent i is assigned a role ρ_i. At round t, agent i receives a local observation o_i^t and maintains a recurrent state h_i^t encoding its interaction history. The global state s^t is available only during centralized training.

Communication is represented by a directed graph $G^t = (\mathcal{I}, \mathcal{E}^t)$, where $(i, j) \in \mathcal{E}^t$ denotes a communication link from agent i to agent j. We initialize G^1 as the complete directed graph without self-loops and set $\mathcal{I}_{\text{comm}}^1 = \mathcal{I}$. At round t, the agents jointly select $\boldsymbol{a}^t = (a_1^t, \dots, a_N^t)$ with $a_i^t \in \mathcal{A}_{\text{comm}}$. The communication-eligible set for the next round is

$$\mathcal{I}_{\text{comm}}^{t+1} = \left\{ i \in \mathcal{I} \mid a_i^t \notin \{a^{\text{off}}, a^{\text{solo}}\} \right\}. \tag{1}$$

The selected modes and directional attention scores then determine the sender and receiver sets $\mathcal{S}_i^{t+1}$ and $\mathcal{R}_i^{t+1}$, which define the directed topology G^{t+1}.

For subsequent analysis, IG (Cover & Thomas, 2006) denotes the topology-induced information gain, which measures the information-acquisition effectiveness of the topology scheduler. Let π denote the topology-scheduling policy and $c_{\text{tok}}^t(\tau)$ the token consumption at round t along trajectory

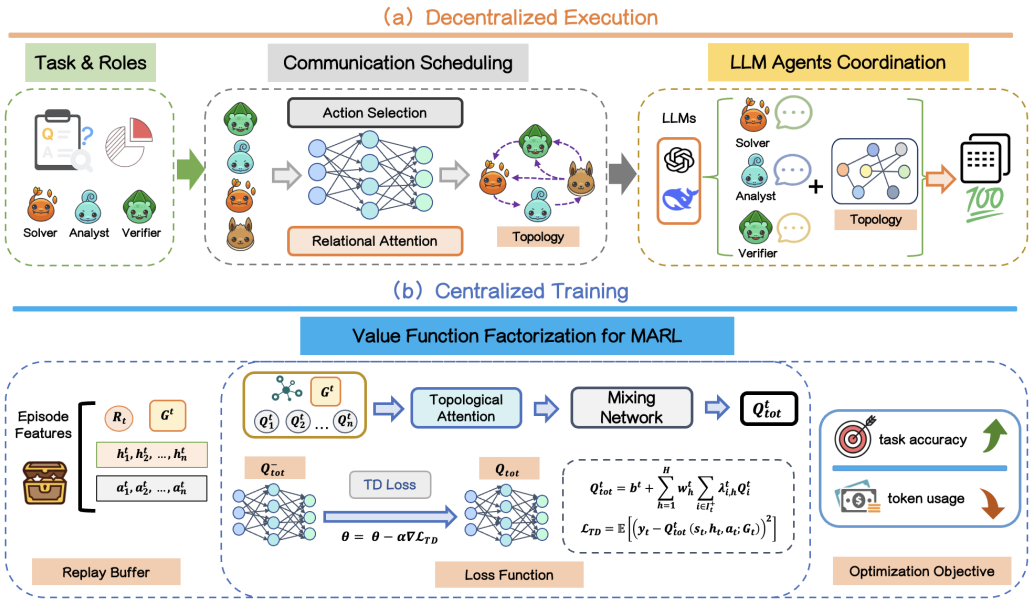

Figure 2: **Overview of ACTS. (a) Decentralized execution.** Local action-value functions select communication modes, while role-aware relational attention selects directed partners and induces the adaptive communication topology. **(b) Centralized training.** Episode transitions are stored in a replay buffer. Topological attention and the mixing network combine the realized topology G^t with local utilities to estimate Q^t_{tot}, which is optimized by TD learning under a task-accuracy and token-cost objective.

τ. The policy is optimized to balance task performance and token efficiency:

$$\pi^\star = \arg\max_{\pi} \mathbb{E}_{\tau\sim\pi}\left[w_{\text{task}} R_{\text{task}}(\tau) - w_{\text{tok}} \min\left(\frac{\sum_{t=1}^{T} c^t_{\text{tok}}(\tau)}{B}, 1\right)\right], \tag{2}$$

where $R_{\text{task}}(\tau)$ denotes the final task score, B is a token-cost normalization scale, and w_{task} and w_{tok} control the trade-off between task performance and token efficiency.

3.2 Relational Attention for Communication Topology Learning

Existing topology schedulers rely on fixed communication patterns and do not explicitly model role-dependent partner relevance (Jiang et al., 2026). ACTS combines communication-mode selection with relational attention over recurrent representations that encode agent roles and evolving local contexts, thereby constructing adaptive directed neighborhoods with a variable number of partners.

At round t, the GNN aggregates each agent's local observation and role information according to the current topology G^t, while a GRU updates the recurrent state using the previous hidden state:

$$h^t_i = \text{GRU}_{\theta_h}\left(\text{GNN}_{\theta_g}\left(o^t_i, e_\rho(\rho_i), G^t\right), h^{t-1}_i\right), \tag{3}$$

where $e_\rho(\rho_i)$ denotes the embedding of agent i's role. The resulting states are processed by a cross-attention module, which computes the directed relevance between each ordered pair of agents:

$$\alpha^t_{ij} = \text{softmax}_{j\neq i}\left(\frac{\left(W_{Q,\text{rel}} h^t_i\right)^\top \left(W_{K,\text{rel}} h^t_j\right)}{\sqrt{d_{\text{rel}}}}\right). \tag{4}$$

The ordered query–key construction produces directional attention weights, with $\alpha^t_{ij} \neq \alpha^t_{ji}$ in general. The cross-attention layer refines each recurrent state by aggregating value projections from other agents with a residual connection, producing the features used for local mode selection:

$$\xi^t_i = h^t_i + \sum_{j\in\mathcal{I}\setminus\{i\}} \alpha^t_{ij} W_{V,\text{rel}} h^t_j, \qquad a^t_i = \arg\max_{a\in\mathcal{A}_{\text{comm}}} Q(\xi^t_i, a). \tag{5}$$

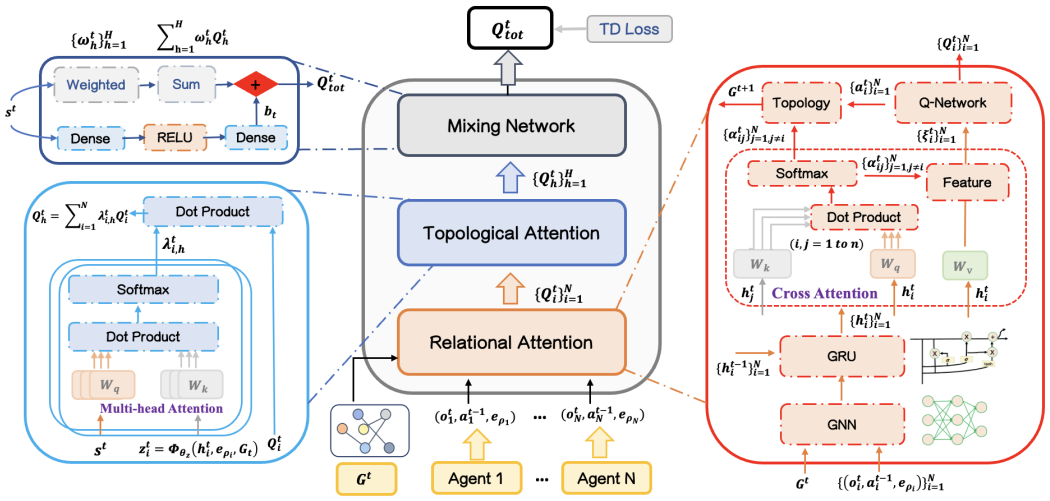

Figure 3: **Architecture of ACTS.** Role-aware relational attention guides directed partner selection. Topological attention weights local utilities according to directed neighborhood representations, and the mixing network constructs Q_{tot}^t. These modules are jointly optimized by minimizing the TD loss within a value function factorization MARL framework under CTDE.

The communication mode specifies the direction and scope of interaction, while relational attention selects the corresponding partners. For the eligible candidate set $\mathcal{J}_i^{t+1}$, the receiver and sender sets are

$$\mathcal{S}_i^{t+1} = \begin{cases} \mathcal{J}_i^{t+1}, & a_i^t = a^{\text{broadcast}}, \\ \{j \in \mathcal{J}_i^{t+1} \mid \alpha_{ij}^t > \bar{\alpha}_i^{\text{out},t}\}, & a_i^t = a^{\text{attn-broadcast}}, \\ \varnothing, & \text{otherwise}, \end{cases}$$
$$\mathcal{R}_i^{t+1} = \begin{cases} \mathcal{J}_i^{t+1}, & a_i^t = a^{\text{receive}}, \\ \{j \in \mathcal{J}_i^{t+1} \mid \alpha_{ji}^t > \bar{\alpha}_i^{\text{in},t}\}, & a_i^t = a^{\text{attn-receive}}, \\ \varnothing, & \text{otherwise}. \end{cases} \tag{6}$$

The resulting directed topology is

$$\mathcal{E}^{t+1} = \left\{(i,j) \in \mathcal{I} \times \mathcal{I} \;\middle|\; i \neq j,\ j \in \mathcal{S}_i^{t+1} \ \vee\ i \in \mathcal{R}_j^{t+1}\right\}, \qquad G^{t+1} = \left(\mathcal{I}, \mathcal{E}^{t+1}\right). \tag{7}$$

By incorporating role embeddings and topology-aware recurrent states, ACTS conditions both action evaluation and partner selection on the current agent context. This decoupled design enables directed, variable-cardinality topologies that adapt across rounds. The complete construction is given in Appendix B.

3.3 Topological Attention for Value Function Factorization

ACTS adopts CTDE value factorization with the Individual-Global-Max property for decentralized greedy execution (Sunehag et al., 2018; Rashid et al., 2018), as reviewed in Appendix A. The QMIX mixer in Agent Q-Mix does not explicitly encode individual directed communication neighborhoods. For a fixed pre-decision context, Appendix C.1 derives the local approximation

$$Q_{\text{tot}}^t(\boldsymbol{a}) \approx b^t + \sum_{i \in \mathcal{I}_{\text{comm}}^t} \beta_i^t Q_i(\xi_i^t, a_i), \tag{8}$$

where β_i^t denotes the local sensitivity to agent i's utility under the global state s^t. Under the graph interaction assumptions, the derivation relates this sensitivity to interactions within the incoming neighborhood $\mathcal{N}_i^{\text{in},t}$ and outgoing neighborhood $\mathcal{N}_i^{\text{out},t}$. This motivates ACTS to learn nonnegative mixing coefficients through topological attention conditioned on the global state and separate representations of incoming and outgoing neighborhoods.

For each candidate communication action a_i, ACTS encodes the directed neighborhoods in G^t by

$$m_{i,r}^t = \text{Agg}_r\left(\left\{\phi_r\left(h_j^t \,\|\, e_\rho(\rho_j)\right) \,\middle|\, j \in \mathcal{N}_i^{r,t}\right\}\right),$$
$$z_i^t = \Phi_{\theta_z}\left(h_i^t \,\|\, e_\rho(\rho_i) \,\|\, m_{i,\text{in}}^t \,\|\, m_{i,\text{out}}^t\right), \qquad r \in \{\text{in}, \text{out}\}. \tag{9}$$

The topology representation z_i^t, used only by the centralized mixer, serves as the key in topological attention, while the global state s^t provides the query. For each head h, we first compute the attention logits and then normalize them over $\mathcal{I}_{\text{comm}}^t$:

$$\eta_{i,h}^t = \frac{\left(W_{Q,\text{top}}^{(h)} s^t\right)^\top \left(W_{K,\text{top}}^{(h)} z_i^t\right)}{\sqrt{d_h}}, \quad \lambda_{i,h}^t = \left[\text{softmax}\left(\left(\eta_{j,h}^t\right)_{j\in\mathcal{I}_{\text{comm}}^t}\right)\right]_i. \tag{10}$$

Thus, $\lambda_{i,h}^t = \exp(\eta_{i,h}^t)/\sum_{j\in\mathcal{I}_{\text{comm}}^t}\exp(\eta_{j,h}^t)$, and the normalization is performed independently for each head.

With nonnegative head weights $w_h^t = |[f_{\theta_w}(s^t)]_h|$ and offset $b^t = V_{\theta_b}(s^t)$, the centralized mixer is

$$Q_{\text{tot}}^t(\boldsymbol{a}) = b^t + \sum_{h=1}^{H} w_h^t \left(\sum_{i\in\mathcal{I}_{\text{comm}}^t} \lambda_{i,h}^t Q_i^t(a_i)\right). \tag{11}$$

Here, $\widehat{\beta}_i^t = \sum_{h=1}^H w_h^t \lambda_{i,h}^t$ is the effective mixing coefficient. Since $w_h^t \geq 0$ and $\lambda_{i,h}^t \geq 0$, $\partial Q_{\text{tot}}^t/\partial Q_i^t(a_i) = \widehat{\beta}_i^t \geq 0$, so the mixer is monotonic in every local action value. With the pre-decision context held fixed, nonnegative mixing ensures that independent greedy actions maximize the learned joint value, as established in Appendix C.3.

$$\prod_{i\in\mathcal{I}_{\text{comm}}^t} \arg\max_{a_i\in\mathcal{A}_i^t} Q_i(\xi_i^t, a_i) \subseteq \arg\max_{\boldsymbol{a}\in\prod_{i\in\mathcal{I}_{\text{comm}}^t}\mathcal{A}_i^t} Q_{\text{tot}}^t(\boldsymbol{a}). \tag{12}$$

The inclusion becomes equality when the attention logits are finite and $\sum_{h=1}^H w_h^t > 0$.

3.4 Cooperative MARL Training and Execution

The training and execution procedures are detailed in Appendix D. During decentralized execution, agent i selects $a_i^t = \arg\max_{a\in\mathcal{A}_{\text{comm}}} Q_i(\xi_i^t, a)$, and the joint action $\boldsymbol{a}^t = (a_1^t, \ldots, a_N^t)$ induces the next-round topology G^{t+1}. During centralized training, the mixer evaluates each replayed transition conditioned on its pre-decision state and topology, with the terminal team reward specified by equation 2.

For each sampled transition, the target action of each communication-eligible agent is selected greedily:

$$a_i^{*,t+1} = \arg\max_{a\in\mathcal{A}_{\text{comm}}} Q_i^-(\xi_i^{t+1}, a), \qquad i \in \mathcal{I}_{\text{comm}}^{t+1}. \tag{13}$$

The TD target and loss are then defined as

$$\begin{aligned} y^t &= r^t + \gamma Q_{\text{tot}}^-\left(\boldsymbol{a}^{*,t+1}\right), \\ \mathcal{L}_{\text{TD}} &= \mathbb{E}_{\mathcal{D}}\left[\left(y^t - Q_{\text{tot}}^t(\boldsymbol{a}^t)\right)^2\right]. \end{aligned} \tag{14}$$

The GNN, GRU, relational-attention module, local action-value networks, topology encoder, and topological mixer are optimized jointly, while the backbone LLM remains frozen. For terminal transitions, the target is $y^t = r^t$. Mixer coefficients are computed from the fixed pre-decision context, and the discrete topology transition is excluded from gradient computation.

4 Experiments

4.1 Experimental Setup

We evaluate ACTS on nine benchmarks spanning four domains: code generation (LiveCodeBench v6 and HumanEval), multiple-choice reasoning (MMLU-Pro and the first 250 questions of Humanity's Last Exam), multi-hop question answering (HotpotQA), and competition mathematics (AIME 2025/2026, HMMT 2025, and Beyond-AIME) (Jain et al., 2024; Chen et al., 2021; Wang et al., 2024;

Table 1: Accuracy (%) across nine benchmarks. Bold marks the best result in each column; ACTS is highlighted in light blue. Avg. is the unweighted macro-average over the nine benchmarks.

Method	LiveCode Bench	Human Eval	HLE	Hotpot QA	MMLU-Pro	AIME 2025	AIME 2026	Beyond-AIME	HMMT 2025	Avg.
LLM backbone: GPT-OSS-120B										
Direct Prompting	88.57	89.63	42.40	20.00	70.00	60.00	86.67	16.00	46.67	57.77
Static topology										
LLM-Debate	92.00	88.41	42.80	21.35	71.43	63.33	86.67	18.00	46.67	58.96
AutoGen	95.43	90.24	41.20	32.03	77.14	70.00	80.00	18.00	46.67	61.19
LangGraph	93.14	89.02	42.40	40.00	84.29	76.67	86.67	23.00	50.00	65.02
Agent Framework	93.14	90.24	42.00	32.03	84.29	73.33	76.67	36.00	40.00	63.08
Centralized adaptive topology										
G-Designer	94.29	93.29	42.80	34.73	70.00	73.33	83.33	28.00	43.33	62.57
TopoDIM	93.71	92.68	42.00	30.68	65.71	66.67	83.33	25.00	33.33	59.23
GTD	94.86	93.90	42.40	35.95	71.43	70.00	80.00	30.00	46.67	62.80
MaAS	94.86	93.29	41.60	33.38	68.57	66.67	86.67	27.00	40.00	61.34
Decentralized learned topology										
Agent Q-Mix	95.43	**100.00**	38.40	40.00	**100.00**	80.00	90.00	36.00	66.67	71.83
ACTS	**96.57**	**100.00**	**43.20**	**46.67**	**100.00**	**83.33**	**93.33**	**42.00**	**70.00**	**75.01**
LLM backbone: DeepSeek-V4-Flash										
Direct Prompting	93.14	98.17	33.60	40.00	72.86	50.00	66.67	10.00	46.67	56.79
Static topology										
LLM-Debate	93.14	96.95	34.00	40.00	61.43	53.33	66.67	12.00	36.67	54.91
AutoGen	**96.00**	98.17	32.80	42.70	71.43	60.00	70.00	15.00	46.67	59.20
LangGraph	92.57	97.56	33.60	46.62	78.57	60.00	70.00	26.00	46.67	61.29
Agent Framework	89.71	98.17	33.20	44.05	75.71	66.67	76.67	20.00	43.33	60.83
Centralized adaptive topology										
G-Designer	93.71	97.56	33.60	46.62	88.57	63.33	73.33	22.00	43.33	62.45
TopoDIM	92.57	97.56	32.80	44.05	88.57	60.00	70.00	18.00	46.67	61.14
GTD	93.71	97.56	34.00	47.97	90.00	63.33	73.33	24.00	43.33	63.03
MaAS	94.29	98.17	32.80	46.62	88.57	63.33	73.33	20.00	43.33	62.27
Decentralized learned topology										
Agent Q-Mix	93.14	**100.00**	32.80	44.00	92.86	60.00	73.33	33.00	40.00	63.24
ACTS	**96.00**	**100.00**	**34.40**	**50.67**	**97.14**	**70.00**	**86.67**	**40.00**	**50.00**	**69.43**

Phan et al., 2025; Yang et al., 2018; Mathematical Association of America, 2025; 2026; MathArena, 2025; ByteDance-Seed, 2025). We use GPT-OSS-120B (OpenAI, 2025) and DeepSeek-V4-Flash (DeepSeek-AI, 2026) as frozen backbone LLMs; each task instantiates three role-conditioned agents and a final answer-aggregation node. Baselines include direct prompting, fixed or framework-based coordination (LLM-Debate, AutoGen, LangGraph, and Agent Framework), task-adaptive topology scheduling (TopoDIM), and MARL-based topology scheduling (Agent Q-Mix) (Du et al., 2023; Wu et al., 2023; LangChain AI, 2024; Sun et al., 2026; Jiang et al., 2026). All methods use the same prompts, team size, communication-round budget, and evaluation protocol. Results are averaged over three independent random seeds. Detailed benchmark, baseline, and training configurations are provided in Appendix E.1.

4.2 Main Results

Accuracy gains from selective communication. Table 1 reports accuracy across all nine benchmarks. ACTS achieves the highest average accuracy on both backbone models, 75.01% on GPT-OSS-120B and 69.43% on DeepSeek-V4-Flash, exceeding Agent Q-Mix by 3.18 and 6.19 points in each case, and attains the best or tied-best result on every benchmark. This advantage is concentrated on tasks where agents hold complementary information, because relational attention selects communication partners by semantic relevance rather than by fixed topology or content-agnostic index rules.

Pareto-optimal efficiency across IG regimes. Figure 4 plots accuracy against token consumption on HotpotQA (IG $>$ 0) and HLE (IG $\approx$ 0); full counts are provided in Table 4 (Appendix E.5). Static topologies broadcast every round regardless of whether communication is useful; centralized adaptive topologies adjust the structure only at generation time, not during execution. Agent Q-

Mix learns agent deactivation and primitive-based routing, but discrete communication primitives provide limited selectivity. ACTS achieves Pareto optimality across all four settings: it matches or exceeds Agent Q-Mix at lower token cost and dominates static and centralized families on both accuracy and cost. Only single-agent prompting uses fewer tokens, at an accuracy cost of approximately 27 percentage points on HotpotQA with GPT-OSS-120B. On HLE, where information gain is near zero, ACTS and single-agent prompting yield comparable performance. By learning whether communication is warranted and routing through attention affinities, ACTS spends tokens only where expected information gain justifies the expenditure.

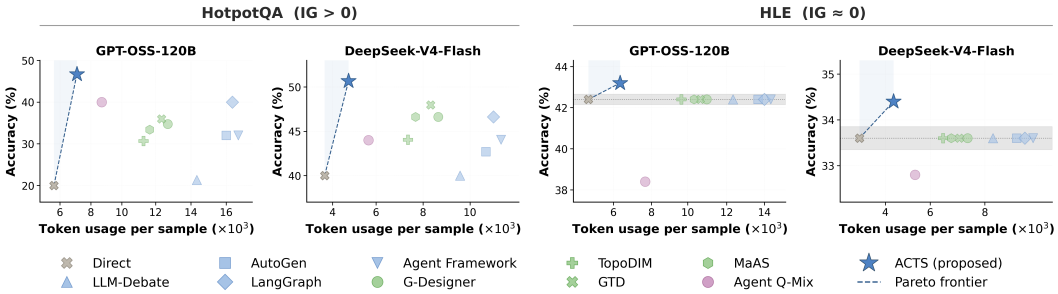

Figure 4: Accuracy vs. token usage for all methods on two representative tasks (HotpotQA with IG > 0 and HLE with IG ≈ 0) and two backbone models. The dashed line traces the Pareto frontier of methods not dominated on both accuracy and cost. ACTS lies on the frontier in all four panels: no other multi-agent method reaches its accuracy at lower or equal cost, and only single-agent prompting is cheaper, at a substantial accuracy loss.

4.3 Experimental Analysis

4.3.1 Information Gain Analysis

Information gain (IG) (Cover & Thomas, 2006) governs whether communication confers an advantage: agents benefit only when they contribute complementary evidence. Figure 5 compares mean accuracy gains across topology families relative to single-agent prompting. Uplift falls below the sampling-noise threshold $x = 2/n$ on HLE and with limited-capacity DeepSeek-V4-Flash on HMMT, yielding null IG. It surpasses this threshold on HotpotQAand contest math, where cross-entity evidence and divergent solution paths engender separable signals, yielding positive IG. ACTS navigates both regimes: attention-driven partner selection amplifies communication when IG is positive; single-agent inference when null.

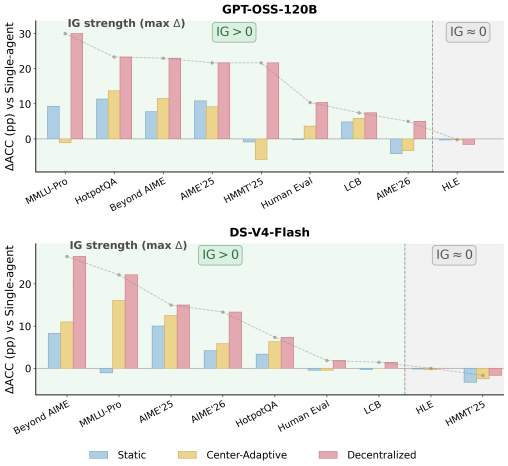

Figure 5: Mean ΔACC of each topology family over single-agent prompting, by backbone model. The dashed line partitions benchmarks into IG > 0 on the left and IG ≈ 0 within $\pm 2/n$ on the right.

4.3.2 Communication Pattern Analysis

ACTS achieves its gains through attention-gated selective communication and dynamic deactivation, two complementary mechanisms analyzed on HotpotQA and HLE, two settings that respectively exhibit high and negligible IG. At each round the agent network computes asymmetric directional scores α_{ij}^t; *attn-broadcast* retains edge $i \to j$ when α_{ij}^t exceeds the mean over eligible candidates, *attn-receive* retains $j \to i$ under the symmetric condition on α_{ji}^t, and together these actions construct sparse neighborhoods specific to each round in place of a fixed topology. Trained end to end with the joint accuracy and token objective, the attention module suppresses redundant interactions, as quantified in Table 4 of Appendix E.5. Independently, the *off* action deactivates an

agent by suppressing its LLM call and outgoing edges; when a majority of agents become inactive the episode terminates early, forgoing rounds unlikely to alter the answer, whereas agents persist when broader collaboration is warranted, and consequently tasks requiring limited collaboration incur cost approaching that of a single agent. Figure 6 contrasts the learned action distributions under DeepSeek-V4-Flash. Under high IG, off stays at 4.2% whereas broadcast reaches 22.7%; under IG near zero, off rises to 21.3%, broadcast and attn-broadcast drop to 12.1% and 6.4%, and attn-receive dominates at 28.3%. Appendix E.4 provides a comparison of the two regimes at the trajectory level.

4.3.3 Ablation Studies

Table 2 isolates the three components distinguishing ACTS from Agent Q-Mix (Jiang et al., 2026). Replacing the topological attention mixer with QMIX (Rashid et al., 2018) induces the strongest degradation, lowering macro-average accuracy from 75.01% to 71.83% on GPT-OSS-120B (OpenAI, 2025) and from 69.43% to 63.24% on DeepSeek-V4-Flash (DeepSeek-AI, 2026), with token usage rising from 5.97 to 7.41 and from 4.03 to 4.96. These results support the effectiveness of the proposed topological attention mixer relative to the QMIX replacement under the evaluated settings. Substituting fixed role-based partners for learned relational routing reduces accuracy to 72.50% and 66.10% and raises token usage to 8.10 and 6.20, the largest overhead among the variants, as fixed assignments cannot adapt partner selection to the task. With role-specific LLM prompts retained, removing role embeddings from relational attention reduces macro-average accuracy to 73.40% and 67.10% and increases token usage to 6.80×10^3 and 4.60×10^3 on GPT-OSS-120B and DeepSeek-V4-Flash, respectively. These results support the contribution of explicit role conditioning to task performance and communication efficiency.

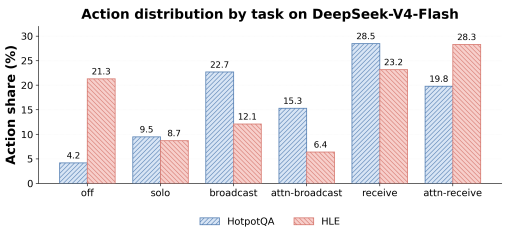

Figure 6: Action distribution by task on DeepSeek-V4-Flash. On HotpotQA, the policy sustains agent activity and broadcast communication; on HLE, it deactivates a substantial fraction of rounds and relies on pull-based retrieval.

Table 2: Ablation results on the two backbone models. $\overline{\text{ACC}}$ and $\overline{\text{Token}}$ denote the macro-average accuracy (%) and macro-average per-sample token usage ($\times 10^3$) across the nine benchmarks. The full ACTS model is highlighted.

	GPT-OSS-120B		DeepSeek-V4-Flash	
Variant	$\overline{\text{ACC}}$	$\overline{\text{Token}}$	$\overline{\text{ACC}}$	$\overline{\text{Token}}$
ACTS	**75.01**	**5.97**	**69.43**	**4.03**
(a) Topological mixer replaced by QMIX	71.83	7.41	63.24	4.96
(b) Learned routing replaced by fixed partners	72.50	8.10	66.10	6.20
(c) Role embeddings removed	73.40	6.80	67.10	4.60

5 Conclusion

We introduced ACTS, a role-aware adaptive communication-topology scheduling framework for efficient LLM-based multi-agent collaboration. Under CTDE, its dual-attention design jointly learns when communication should occur and which directed agent pairs should exchange information: role-aware relational attention selects variable-cardinality partners from agent roles and evolving execution states, while a topological attention mixer incorporates the realized directed topology into team-value factorization for topology-aware credit assignment. Across nine benchmarks and two frozen backbone LLMs, ACTS improves the accuracy–efficiency trade-off over Agent Q-Mix by up to 6.19 percentage points in accuracy and up to 15.42% in token reduction. These results demonstrate the benefit of learning communication modes and directed partner relations jointly for efficient role-aware LLM collaboration.

6 AI USE STATEMENT

Generative AI tools were used to assist with literature search and reference verification, drafting and revision of manuscript text, language editing, and LaTeX formatting. They were not used to collect or generate data, run experiments, modify empirical results, or determine the final scientific claims. Other required disclosure categories are not applicable to this work.

All AI-assisted material was reviewed and revised by the authors. The authors independently checked the manuscript text, equations, proofs, citations, bibliographic links, reported values, and LaTeX compilation. The final experimental results and conclusions were verified against the corresponding experimental records. The authors take full responsibility for the final content of this work, including all text, claims, references, code, and other artifacts.

REFERENCES

ByteDance-Seed. BeyondAIME: Advancing math reasoning evaluation beyond high school olympiads. Hugging Face Datasets, 2025. URL https://huggingface.co/datasets/ByteDance-Seed/BeyondAIME.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. AgentVerse: Facilitating multi-agent collaboration and exploring emergent behaviors, 2023. URL https://arxiv.org/abs/2308.10848.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2 edition, 2006. doi: 10.1002/047174882X. URL https://books.google.com/books?id=j0DBDwAAQBAJ.

DeepSeek-AI. Deepseek-v4-flash. Official model documentation, 2026. URL https://api-docs.deepseek.com/.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate, 2023. URL https://arxiv.org/abs/2305.14325.

Jakob N. Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://papers.nips.cc/paper_files/paper/2016/hash/c7635bfd99248a2cdef8249ef7bfbef4-Abstract.html.

Sirui Hong, Mingchen Zhuge, Jonathan 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, 2023. URL https://arxiv.org/abs/2308.00352v5.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024. URL https://arxiv.org/abs/2403.07974.

Eric Hanchen Jiang, Mengting Li, Guancheng Wan, Sophia Yin, Yuchen Wu, Xiao Liang, Xinfeng Li, Yizhou Sun, Wei Wang, Kai-Wei Chang, and Ying Nian Wu. Dynamic generation of multi-LLM agents communication topologies with graph diffusion models, 2025. URL https://arxiv.org/abs/2510.07799.

Eric Hanchen Jiang, Levina Li, Rui Sun, Xiao Liang, Yubei Li, Yuchen Wu, Haozheng Luo, Hengli Li, Zhi Zhang, Zhaolu Kang, Kai-Wei Chang, and Ying Nian Wu. Agent Q-Mix: Selecting the right action for LLM multi-agent systems through reinforcement learning, 2026. URL https://arxiv.org/abs/2604.00344.

LangChain AI. LangGraph. Software repository, 2024. URL https://github.com/langchain-ai/langgraph.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a3621ee907def47c1b952ade25c67698-Abstract-Conference.html.

Shiyuan Li, Yixin Liu, Qingsong Wen, Chengqi Zhang, and Shirui Pan. Assemble your crew: Automatic multi-agent communication topology design via autoregressive graph generation, 2025. URL https://arxiv.org/abs/2507.18224.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. A dynamic LLM-powered agent network for task-oriented agent collaboration. In *Proceedings of the First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=XII0Wp1XA9.

MathArena. HMMT february 2025 competition problems. Hugging Face Datasets, 2025. URL https://huggingface.co/datasets/MathArena/hmmt_feb_2025.

Mathematical Association of America. 2025 AIME I. AoPS Wiki, 2025. URL https://artofproblemsolving.com/wiki/index.php/2025_AIME_I.

Mathematical Association of America. 2026 AIME I. AoPS Wiki, 2026. URL https://artofproblemsolving.com/wiki/index.php/2026_AIME_I.

Microsoft. Microsoft Agent Framework. Software repository, 2025. URL https://github.com/microsoft/agent-framework.

Frans A. Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*. Springer, 2016. doi: 10.1007/978-3-319-28929-8. URL https://link.springer.com/book/10.1007/978-3-319-28929-8.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025. URL https://arxiv.org/abs/2508.10925.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025. URL https://arxiv.org/abs/2501.14249.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15174–15186. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.810. URL https://aclanthology.org/2024.acl-long.810/.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob N. Foerster, and Shimon Whiteson. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4295–4304. PMLR, 2018. URL https://proceedings.mlr.press/v80/rashid18a.html.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5887–5896. PMLR, 2019. URL https://proceedings.mlr.press/v97/son19a.html.

Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://papers.nips.cc/paper_files/paper/2016/hash/55b1927fdafef39c48e5b73b5d61ea60-Abstract.html.

Rui Sun, Jie Ding, Chenghua Gong, Tianjun Gu, Yihang Jiang, Juyuan Zhang, Liming Pan, and Linyuan Lü. TopoDIM: One-shot topology generation of diverse interaction modes for multi-agent systems, 2026. URL https://arxiv.org/abs/2601.10120.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2085–2087, 2018. URL https://www.ifaamas.org/Proceedings/aamas2018/pdfs/p2085.pdf.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. QPLEX: Duplex dueling multi-agent q-learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Rcmk0xxIQV.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3018. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/ad236edc564f3e3156e1b2feafb99a24-Abstract.html.

Zhexuan Wang, Yutong Wang, Xuebo Liu, Liang Ding, Miao Zhang, Jie Liu, and Min Zhang. AgentDropout: Dynamic agent elimination for token-efficient and high-performance LLM-based multi-agent collaboration, 2025. URL https://arxiv.org/abs/2503.18891.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang 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://arxiv.org/abs/2308.08155.

Bingyu Yan, Zhibo Zhou, Litian Zhang, Lian Zhang, Ziyi Zhou, Dezhuang Miao, Zhoujun Li, Chaozhuo Li, and Xiaoming Zhang. Beyond self-talk: A communication-centric survey of llm-based multi-agent systems. *arXiv preprint arXiv:2502.14321*, 2025. doi: 10.48550/arXiv.2502.14321. URL https://arxiv.org/abs/2502.14321.

Yaodong Yang, Jianye Hao, Ben Liao, Kun Shao, Guangyong Chen, Wulong Liu, and Hongyao Tang. Qatten: A general framework for cooperative multiagent reinforcement learning. *arXiv preprint arXiv:2002.03939*, 2020. URL https://arxiv.org/abs/2002.03939.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. G-Designer: Architecting multi-agent communication topologies via graph neural networks, 2024. URL https://arxiv.org/abs/2410.11782.

Guibin Zhang, Luyang Niu, Junfeng Fang, Kun Wang, Lei Bai, and Xiang Wang. Multi-agent architecture search via agentic supernet, 2025. URL https://arxiv.org/abs/2502.04180.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. GPTSwarm: Language agents as optimizable graphs. In *Proceedings of the 41st*

International Conference on Machine Learning, volume 235 of *Proceedings of Machine Learning Research*, pp. 62743–62767. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhuge24a.html.

A Value Function Factorization in Cooperative Multi-Agent Reinforcement Learning

A.1 Centralized Training with Decentralized Execution

Cooperative multi-agent reinforcement learning (MARL) considers a set of agents that jointly maximize a shared return. Let s^t denote the global state, τ_i^t the local action-observation history of agent i, and $\boldsymbol{u}^t = (u_1^t, \ldots, u_N^t)$ the joint action. The objective is

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=1}^{T} \gamma^{t-1} r^t \right], \tag{15}$$

where r^t is the shared reward and $\gamma \in [0, 1)$ is the discount factor.

In partially observable environments, each agent makes decisions from its local action-observation history. Independent value-based learning is challenging because the joint action space grows exponentially with the number of agents, while the effective environment of each agent changes as the other agents update their policies. Centralized training with decentralized execution (CTDE) addresses these issues by allowing the training procedure to access centralized information, while restricting execution to local histories.

Let $Q_i(\tau_i^t, u_i)$ denote the individual action-value function of agent i. Under CTDE, the centralized joint action-value function can be represented as

$$Q_{\text{tot}}\left(s^t, \boldsymbol{\tau}^t, \boldsymbol{u}^t\right) = F_\psi \left(s^t, \left\{Q_i(\tau_i^t, u_i^t)\right\}_{i=1}^N\right), \tag{16}$$

where F_ψ is a centralized factorization function. This formulation allows centralized training to model inter-agent dependencies while retaining decentralized action selection at execution time.

A.2 Value Function Factorization

Value Decomposition Networks (VDN) use the additive factorization

$$Q_{\text{tot}}\left(\boldsymbol{\tau}, \boldsymbol{u}\right) = \sum_{i=1}^{N} Q_i(\tau_i, u_i), \tag{17}$$

which supports decentralized execution but cannot represent general non-additive interactions among agents (Sunehag et al., 2018).

QMIX introduces a nonlinear factorization function (Rashid et al., 2018):

$$Q_{\text{tot}} = F_\psi \left(Q_1, \ldots, Q_N; s\right), \tag{18}$$

subject to the monotonicity condition

$$\frac{\partial Q_{\text{tot}}}{\partial Q_i} \geq 0, \qquad i = 1, \ldots, N. \tag{19}$$

Thus, increasing an individual utility cannot decrease the joint value when the remaining utilities and the centralized state are fixed. QMIX typically enforces this condition by generating nonnegative factorization weights from the global state through hypernetworks.

Subsequent methods improve the expressiveness of value function factorization in different ways. QTRAN introduces transformation and consistency constraints to represent a broader class of joint value functions (Son et al., 2019). QPLEX uses a duplex dueling architecture to construct a more expressive value–advantage factorization while preserving decentralized consistency (Wang et al., 2021). Attention- based factorization methods assign context-dependent weights to individual utilities, allowing their contributions to vary with the centralized context (Yang et al., 2020).

A.3 Monotonic Value Mixing and the IGM Property

A factorized joint action-value function satisfies the Individual–Global–Max (IGM) (Sunehag et al., 2018; Rashid et al., 2018; Son et al., 2019) property if its globally greedy action is consistent with the individual greedy actions:

$$\arg\max_{\boldsymbol{u}} Q_{\mathrm{tot}}\left(s, \boldsymbol{\tau}, \boldsymbol{u}\right) = \left(\arg\max_{u_1} Q_1(\tau_1, u_1), \ldots, \arg\max_{u_N} Q_N(\tau_N, u_N)\right). \tag{20}$$

With consistent tie-breaking, this property means that the globally optimal joint action can be recovered by independently maximizing the individual action-value functions.

Componentwise monotonicity is a sufficient condition for IGM. For fixed s and $\boldsymbol{\tau}$, consider two joint actions that differ only in agent i's action. If one action produces a larger value of $Q_i(\tau_i, u_i)$, the monotonicity condition ensures that it cannot produce a smaller value of Q_{tot}. Applying this argument coordinate-wise to all agents shows that independent greedy action selection yields a globally optimal joint action.

A.4 Temporal-Difference Optimization

Cooperative value function factorization methods are commonly trained with a centralized temporal-difference objective while retaining decentralized execution (Sunehag et al., 2018; Rashid et al., 2018; Son et al., 2019). For a transition $(s^t, \boldsymbol{\tau}^t, \boldsymbol{u}^t, r^t, s^{t+1}, d^t)$, where d^t indicates episode termination, a target network forms

$$y^t = r^t + \gamma(1 - d^t) Q^-_{\mathrm{tot}}\left(s^{t+1}, \boldsymbol{\tau}^{t+1}, \boldsymbol{u}^{*,t+1}\right). \tag{21}$$

When the factorization satisfies the Individual–Global-Max property, the target joint action can be selected independently (Rashid et al., 2018; Wang et al., 2021):

$$u_i^{*,t+1} = \arg\max_{u_i} Q_i^-\left(\tau_i^{t+1}, u_i\right), \qquad i = 1, \ldots, N. \tag{22}$$

The online parameters are optimized by minimizing

$$\mathcal{L}_{\mathrm{TD}} = \mathbb{E}_{\mathcal{D}}\left[\left(y^t - Q_{\mathrm{tot}}\left(s^t, \boldsymbol{\tau}^t, \boldsymbol{u}^t\right)\right)^2\right]. \tag{23}$$

Experience replay and delayed target networks are commonly used to stabilize optimization under deep function approximation (Rashid et al., 2018; Son et al., 2019).

B Construction of the Directed Communication Topology

This section gives the complete construction of $G^{t+1} = \Psi(\boldsymbol{a}^t, \boldsymbol{\alpha}^t)$. After communication-mode selection, define the communication-eligible agents and candidate partners as

$$\mathcal{I}_{t+1}^{\mathrm{comm}} = \left\{i \in \mathcal{I} \;\middle|\; a_i^t \notin \left\{a^{\mathrm{off}}, a^{\mathrm{solo}}\right\}\right\}, \qquad \mathcal{J}_i^{t+1} = \mathcal{I}_{t+1}^{\mathrm{comm}} \setminus \{i\}. \tag{24}$$

The index $t+1$ indicates that these sets determine the topology produced by the decisions at round t.

For $i \in \mathcal{I}_{t+1}^{\mathrm{comm}}$ with $\mathcal{J}_i^{t+1} \neq \varnothing$, define the outgoing and incoming mean scores as

$$\bar{\alpha}_i^{\mathrm{out},t} = \frac{1}{|\mathcal{J}_i^{t+1}|} \sum_{j \in \mathcal{J}_i^{t+1}} \alpha_{ij}^t, \qquad \bar{\alpha}_i^{\mathrm{in},t} = \frac{1}{|\mathcal{J}_i^{t+1}|} \sum_{j \in \mathcal{J}_i^{t+1}} \alpha_{ji}^t. \tag{25}$$

The requested receiver and sender sets are

$$\mathcal{S}_i^{t+1} = \begin{cases} \mathcal{J}_i^{t+1}, & a_i^t = a^{\mathrm{broadcast}}, \\ \left\{j \in \mathcal{J}_i^{t+1} \;\middle|\; \alpha_{ij}^t > \bar{\alpha}_i^{\mathrm{out},t}\right\}, & a_i^t = a^{\mathrm{attn\text{-}broadcast}}, \\ \varnothing, & \text{otherwise}, \end{cases}$$

$$\mathcal{R}_i^{t+1} = \begin{cases} \mathcal{J}_i^{t+1}, & a_i^t = a^{\mathrm{receive}}, \\ \left\{j \in \mathcal{J}_i^{t+1} \;\middle|\; \alpha_{ji}^t > \bar{\alpha}_i^{\mathrm{in},t}\right\}, & a_i^t = a^{\mathrm{attn\text{-}receive}}, \\ \varnothing, & \text{otherwise}. \end{cases} \tag{26}$$

When $\mathcal{J}_i^{t+1} = \varnothing$, both request sets are defined as empty.

The directed edge set is then

$$\mathcal{E}^{t+1} = \left\{(i,j) \in \mathcal{I}_{t+1}^{\text{comm}} \times \mathcal{I}_{t+1}^{\text{comm}} \;\middle|\; i \neq j,\ j \in \mathcal{S}_i^{t+1} \vee i \in \mathcal{R}_j^{t+1}\right\}, \tag{27}$$

and

$$G^{t+1} = \left(\mathcal{I}, \mathcal{E}^{t+1}\right). \tag{28}$$

Thus, $a^{\text{broadcast}}$ and a^{receive} request all eligible interactions in the corresponding direction, whereas $a^{\text{attn-broadcast}}$ and $a^{\text{attn-receive}}$ retain only interactions whose scores exceed the corresponding mean. The strict mean threshold allows the number of selected partners to adapt to the score distribution without introducing a fixed top-k hyperparameter.

C Theoretical Analysis of Topological Value Factorization

At a non-terminal round t, let $\mathcal{I}_{\text{comm}}^t \neq \varnothing$ denote the set of agents eligible to make communication decisions, fixed before the current actions are selected. The corresponding pre-decision context is

$$\mathcal{C}^t = \left(s^t,\ G^t,\ \mathcal{I}_{\text{comm}}^t,\ \{\xi_i^t, z_i^t\}_{i \in \mathcal{I}_{\text{comm}}^t}\right), \tag{29}$$

where ξ_i^t is the input to the individual action-value function and z_i^t encodes agent i's directed neighborhoods in G^t for centralized value factorization. Let $\boldsymbol{a}^t = (a_i^t)_{i \in \mathcal{I}}$ denote the full joint action, with $a_i^t = a^{\text{off}}$ for every $i \notin \mathcal{I}_{\text{comm}}^t$. The selected actions determine

$$\mathcal{I}_{\text{comm}}^{t+1} = \left\{i \in \mathcal{I} :\ a_i^t \notin \{a^{\text{off}}, a^{\text{solo}}\}\right\}, \qquad G^{t+1} = \Psi\left(\mathcal{C}^t, \boldsymbol{a}^t\right), \tag{30}$$

where Ψ denotes the construction of Appendix B. Because every agent outside $\mathcal{I}_{\text{comm}}^t$ is assigned a^{off}, Equation 30 implies $\mathcal{I}_{\text{comm}}^{t+1} \subseteq \mathcal{I}_{\text{comm}}^t$: the eligible set contracts monotonically along an episode, so an agent that selects a^{off} or a^{solo} withdraws from communication for the remainder of the episode.

C.1 Local Expansion and Interaction-Dependent Coefficients

Fix $\mathcal{C}^t$ and let $\boldsymbol{q} = (q_i)_{i \in \mathcal{I}_{\text{comm}}^t}$ denote a vector of individual utility values. Suppose a reference joint action-value function is represented locally by a map

$$F_{\mathcal{C}^t} : \mathcal{U} \to \mathbb{R}, \qquad \mathcal{U} \subseteq \mathbb{R}^{|\mathcal{I}_{\text{comm}}^t|} \text{ open}, \qquad F_{\mathcal{C}^t} \in C^2(\mathcal{U}).$$

This smoothness assumption is imposed in utility space and therefore does not require the discrete communication actions to be differentiable. For $\boldsymbol{u} \in \mathcal{U}$ define the local sensitivity

$$\beta_i^{\text{ref},t}(\boldsymbol{u}) = \partial_i F_{\mathcal{C}^t}(\boldsymbol{u}), \tag{31}$$

and, for a reference point $\bar{\boldsymbol{q}} \in \mathcal{U}$, write

$$\beta_i^{\text{ref},t} = \beta_i^{\text{ref},t}(\bar{\boldsymbol{q}}), \qquad c^{\text{ref},t} = F_{\mathcal{C}^t}(\bar{\boldsymbol{q}}) - \sum_{i \in \mathcal{I}_{\text{comm}}^t} \beta_i^{\text{ref},t} \bar{q}_i. \tag{32}$$

Proposition C.1 (Local weighted expansion)**.** *If the line segment between $\bar{\boldsymbol{q}}$ and $\boldsymbol{q}$ lies in $\mathcal{U}$, then*

$$F_{\mathcal{C}^t}(\boldsymbol{q}) = c^{\text{ref},t} + \sum_{i \in \mathcal{I}_{\text{comm}}^t} \beta_i^{\text{ref},t} q_i + R_2^t. \tag{33}$$

If in addition $\|\nabla^2 F_{\mathcal{C}^t}(\boldsymbol{u})\|_{2\to 2} \leq L$ along that segment, then

$$\left|R_2^t\right| \leq \frac{L}{2} \left\|\boldsymbol{q} - \bar{\boldsymbol{q}}\right\|_2^2. \tag{34}$$

Proof. Let $\boldsymbol{\Delta} = \boldsymbol{q} - \bar{\boldsymbol{q}}$ and $g(u) = F_{\mathcal{C}^t}(\bar{\boldsymbol{q}} + u\boldsymbol{\Delta})$. Since $g \in C^2([0,1])$, the integral form of Taylor's theorem gives

$$\begin{aligned} F_{\mathcal{C}^t}(\boldsymbol{q}) = F_{\mathcal{C}^t}(\bar{\boldsymbol{q}}) + \nabla F_{\mathcal{C}^t}(\bar{\boldsymbol{q}})^\top \boldsymbol{\Delta} \\ + \int_0^1 (1-u)\, \boldsymbol{\Delta}^\top \nabla^2 F_{\mathcal{C}^t}(\bar{\boldsymbol{q}} + u\boldsymbol{\Delta}) \boldsymbol{\Delta}\, du. \end{aligned} \tag{35}$$

Because $\Delta_i = q_i - \bar{q}_i$ and $\beta_i^{\mathrm{ref},t} = \partial_i F_{\mathcal{C}^t}(\bar{\boldsymbol{q}})$, the first two terms of Equation 35 equal $c^{\mathrm{ref},t} + \sum_i \beta_i^{\mathrm{ref},t} q_i$ by Equation 32, which yields Equation 33 with R_2^t the integral term. For the remainder, $|\boldsymbol{\Delta}^\top \nabla^2 F_{\mathcal{C}^t} \boldsymbol{\Delta}| \le \|\nabla^2 F_{\mathcal{C}^t}\|_{2\to 2} \|\boldsymbol{\Delta}\|_2^2 \le L\|\boldsymbol{\Delta}\|_2^2$, and $\int_0^1 (1-u)\, du = \frac{1}{2}$ gives Equation 34. □

Remark C.2. *Proposition C.1 is a statement about the* form *of a local approximation rather than a claim of novelty: every C^2 reference value admits weighted-sum coefficients equal to its local sensitivities. Its role here is to identify what a mixer must supply, namely per-agent coefficients that vary with the pre-decision context, and to bound the error of the weighted-sum form by the curvature of $F_{\mathcal{C}^t}$. The analysis that follows concerns* which *quantities those coefficients depend on.*

The coefficient $\beta_i^{\mathrm{ref},t}$ is the local sensitivity of the reference joint value to agent i's utility, and its variation across operating points is governed by mixed derivatives. Applying the fundamental theorem of calculus to $u \mapsto \partial_i F_{\mathcal{C}^t}(\bar{\boldsymbol{q}} + u\boldsymbol{\delta})$ and using the chain rule gives, for any displacement $\boldsymbol{\delta}$ whose segment lies in $\mathcal{U}$,

$$\beta_i^{\mathrm{ref},t}(\bar{\boldsymbol{q}} + \boldsymbol{\delta}) - \beta_i^{\mathrm{ref},t}(\bar{\boldsymbol{q}}) = \sum_{j \in \mathcal{I}_{\mathrm{comm}}^t} \delta_j \int_0^1 \partial_{ij} F_{\mathcal{C}^t}(\bar{\boldsymbol{q}} + u\boldsymbol{\delta})\, du. \tag{36}$$

For $j \neq i$, the mixed derivative $\partial_{ij} F_{\mathcal{C}^t}$ characterizes how agent j's utility affects the joint value's local sensitivity to agent i's utility. This dependence motivates context-dependent mixing coefficients, consistent with attention-based value factorization (Yang et al., 2020).

A direct connection to the communication neighborhoods follows once the reference value is graph structured. Let

$$\mathcal{E}_{\mathrm{comm}}^t = \mathcal{E}^t \cap \big(\mathcal{I}_{\mathrm{comm}}^t \times \mathcal{I}_{\mathrm{comm}}^t\big), \quad \mathcal{N}_i^{\mathrm{out},t} = \{j : (i,j) \in \mathcal{E}_{\mathrm{comm}}^t\}, \quad \mathcal{N}_i^{\mathrm{in},t} = \{j : (j,i) \in \mathcal{E}_{\mathrm{comm}}^t\} \tag{37}$$

denote the directed, self-loop-free edge set induced on $\mathcal{I}_{\mathrm{comm}}^t$ and the associated neighborhoods. Agents that exchange no messages at round t influence one another only through the shared team reward, so modelling the interaction terms of the reference value as supported on $\mathcal{E}_{\mathrm{comm}}^t$ is the natural structural assumption.

Proposition C.3 (Graph-supported coefficients)**.** *Let $\mathcal{S}(G^t)$ be a collection of scopes $S \subseteq \mathcal{I}_{\mathrm{comm}}^t$ supported on G^t, write $\boldsymbol{q}_S = (q_j)_{j \in S}$, and suppose*

$$F_{\mathcal{C}^t}(\boldsymbol{q}) = a_{\mathcal{C}^t} + \sum_{S \in \mathcal{S}(G^t)} f_S(\boldsymbol{q}_S; G^t), \tag{38}$$

with every f_S continuously differentiable in its arguments. Then

$$\beta_i^{\mathrm{ref},t}(\boldsymbol{q}) = \sum_{\substack{S \in \mathcal{S}(G^t) \\ i \in S}} \partial_i f_S(\boldsymbol{q}_S; G^t). \tag{39}$$

Proof. Differentiating Equation 38 with respect to q_i annihilates $a_{\mathcal{C}^t}$ and every factor whose scope does not contain i, since such factors do not depend on q_i. Summing the derivatives of the remaining factors gives Equation 39. □

Proposition C.3 places no restriction on the cardinality of the scopes, so every graph-induced interaction factor containing agent i, including higher-order ones, can contribute to its coefficient. The pairwise case isolates the two directed contributions.

Corollary C.4 (Directed-neighborhood dependence)**.** *Suppose*

$$F_{\mathcal{C}^t}(\boldsymbol{q}) = a_{\mathcal{C}^t} + \sum_{i \in \mathcal{I}_{\mathrm{comm}}^t} f_i(q_i) + \sum_{(i,j) \in \mathcal{E}_{\mathrm{comm}}^t} f_{ij}(q_i, q_j). \tag{40}$$

Then

$$\begin{aligned} \beta_i^{\mathrm{ref},t}(\boldsymbol{q}) = f_i'(q_i) &+ \sum_{j \in \mathcal{N}_i^{\mathrm{out},t}} \partial_1 f_{ij}(q_i, q_j) \\ &+ \sum_{j \in \mathcal{N}_i^{\mathrm{in},t}} \partial_2 f_{ji}(q_j, q_i). \end{aligned} \tag{41}$$

Proof. Apply Proposition C.3 with $\mathcal{S}(G^t) = \{\{i\} : i \in \mathcal{I}^t_{\mathrm{comm}}\} \cup \mathcal{E}^t_{\mathrm{comm}}$. The unary scope containing i contributes $f_i'(q_i)$. Because $\mathcal{E}^t_{\mathrm{comm}}$ is directed and self-loop free, a pairwise scope contains i in exactly one of two ways: for an outgoing edge (i, j) the utility q_i occupies the first argument of f_{ij}, and for an incoming edge (j, i) it occupies the second argument of f_{ji}. Summing the corresponding partial derivatives gives Equation 41. □

Corollary C.4 is the design justification for the topology encoder of Equation 9. The incoming and outgoing neighborhoods enter $\beta_i^{\mathrm{ref},t}$ through distinct partial derivatives, $\partial_2 f_{ji}$ and $\partial_1 f_{ij}$ respectively, so a representation that merges them into a single undirected neighborhood conflates two terms that are not interchangeable. A representation built from graph-level statistics such as edge count or density is weaker still: it discards the identity of agent i and therefore cannot express the restriction to scopes containing i that Equation 39 imposes. `ACTS` accordingly encodes both directed neighborhoods in z_i^t. It does not evaluate $\beta_i^{\mathrm{ref},t}$ analytically; it learns a nonnegative surrogate $\widehat{\beta}_i^t$ from TD targets, as described next.

C.2 Topology-Aware TD Credit Assignment

For $i \in \mathcal{I}^t_{\mathrm{comm}}$ define the individual utility and the topological-attention logit of head h,

$$Q_i^t(a_i) = Q_i(\xi_i^t, a_i), \qquad \eta_{i,h}^t = \frac{\left(W^{(h)}_{Q,\mathrm{top}} s^t\right)^\top \left(W^{(h)}_{K,\mathrm{top}} z_i^t\right)}{\sqrt{d_h}}, \tag{42}$$

with weights normalized over the eligible agents,

$$\lambda_{i,h}^t = \frac{\exp\left(\eta_{i,h}^t\right)}{\sum_{j \in \mathcal{I}^t_{\mathrm{comm}}} \exp\left(\eta_{j,h}^t\right)}. \tag{43}$$

Let $w_h^t = |[f_{\theta_w}(s^t)]_h| \geq 0$ be the nonnegative head weights and $b^t = V_{\theta_b}(s^t)$ the state-dependent bias. The head-level and joint action values are

$$\begin{aligned} Q_h^t(\boldsymbol{a}) &= \sum_{i \in \mathcal{I}^t_{\mathrm{comm}}} \lambda_{i,h}^t Q_i^t(a_i), \\ Q_{\mathrm{tot}}^t(\mathcal{C}^t, \boldsymbol{a}) &= b^t + \sum_{h=1}^{H} w_h^t Q_h^t(\boldsymbol{a}) = b^t + \sum_{i \in \mathcal{I}^t_{\mathrm{comm}}} \widehat{\beta}_i^t Q_i^t(a_i), \end{aligned} \tag{44}$$

where the effective mixing coefficient is

$$\widehat{\beta}_i^t = \sum_{h=1}^{H} w_h^t \lambda_{i,h}^t \;\geq\; 0. \tag{45}$$

The second equality in Equation 44 follows by exchanging the two finite sums, and shows that the multi-head mixer is, for a fixed context, a nonnegatively weighted sum of individual utilities whose weights are determined by $\{z_i^t\}$ and hence by G^t.

Remark C.5 (Functional form of a head)**.** *Since s^t is shared by all agents within a head, Equation 42 can be written as $\eta_{i,h}^t = \langle v_h(s^t), z_i^t \rangle$ with $v_h(s^t) = (W^{(h)}_{K,\mathrm{top}})^\top W^{(h)}_{Q,\mathrm{top}} s^t / \sqrt{d_h}$. Each head therefore contributes one state-dependent linear direction in neighborhood-embedding space along which agents are compared, and w_h^t determines how much that direction matters in the current state.*

For a sampled transition let y^t denote the detached TD target and set

$$\delta^t = Q_{\mathrm{tot}}^t(\mathcal{C}^t, \boldsymbol{a}^t) - y^t, \qquad \ell_{\mathrm{TD}}^t = \tfrac{1}{2}\left(\delta^t\right)^2. \tag{46}$$

Proposition C.6 (Topology-aware TD gradients)**.** *Conditional on the context $\mathcal{C}^t$ and with y^t treated as a constant,*

$$\frac{\partial \ell_{\mathrm{TD}}^t}{\partial Q_i^t(a_i^t)} = \delta^t \widehat{\beta}_i^t, \qquad \frac{\partial \ell_{\mathrm{TD}}^t}{\partial w_h^t} = \delta^t Q_h^t(\boldsymbol{a}^t), \tag{47}$$

and for every eligible agent i and head h,

$$\frac{\partial \ell_{\mathrm{TD}}^t}{\partial \eta_{i,h}^t} = \delta^t w_h^t \lambda_{i,h}^t \left(Q_i^t(a_i^t) - Q_h^t(\boldsymbol{a}^t)\right). \tag{48}$$

Proof. equation 44 gives $\partial Q^t_{\mathrm{tot}}/\partial Q^t_i(a^t_i) = \widehat{\beta}^t_i$ and $\partial Q^t_{\mathrm{tot}}/\partial w^t_h = Q^t_h(\boldsymbol{a}^t)$; since $\partial \ell^t_{\mathrm{TD}}/\partial Q^t_{\mathrm{tot}} = \delta^t$ when y^t is detached, the chain rule yields Equation 47. For the logits, the softmax Jacobian of Equation 43 is

$$\frac{\partial \lambda^t_{j,h}}{\partial \eta^t_{i,h}} = \lambda^t_{j,h}\left(\mathbb{I}[i=j] - \lambda^t_{i,h}\right),$$

and $\eta^t_{i,h}$ enters only head h, so

$$\frac{\partial Q^t_h(\boldsymbol{a}^t)}{\partial \eta^t_{i,h}} = \sum_{j \in \mathcal{I}^t_{\mathrm{comm}}} Q^t_j(a^t_j)\lambda^t_{j,h}\left(\mathbb{I}[i=j] - \lambda^t_{i,h}\right) = \lambda^t_{i,h}\left(Q^t_i(a^t_i) - Q^t_h(\boldsymbol{a}^t)\right),$$

where the last step uses $\sum_j \lambda^t_{j,h} Q^t_j(a^t_j) = Q^t_h(\boldsymbol{a}^t)$. Combining with $\partial Q^t_{\mathrm{tot}}/\partial Q^t_h = w^t_h$ proves Equation 48. □

equation 47 shows that $\widehat{\beta}^t_i$ is simultaneously the Jacobian of the mixer with respect to agent i's selected utility and the scale of the direct TD signal that this utility receives: monotonicity and credit assignment are governed by the same quantity. Because $\widehat{\beta}^t_i$ depends on z^t_i, and z^t_i encodes the directed neighborhoods of G^t by Corollary C.4, the allocation is topology aware. equation 48 has the form of an advantage: the deviation $Q^t_i - Q^t_h$ compares agent i's utility against the weighted mean of head h, so a head increases its weight on agents whose utility exceeds that mean whenever the joint value must increase ($\delta^t < 0$), and decreases it otherwise. The head weights are produced by an absolute value and are therefore differentiable except on the measure-zero set $[f_{\theta_w}(s^t)]_h = 0$, where the subgradient is taken to be zero, as is standard for monotonic mixers (Rashid et al., 2018).

Remark C.7 (Where credit for partner selection enters)**.** *All quantities in Equation 44 are computed from the pre-decision context, so Q^t_{tot} does not depend on the topology G^{t+1} that $\boldsymbol{a}^t$ induces, and the discrete map Ψ of Equation 30 is never differentiated. Credit for a partner configuration is consequently not delivered within round t; it is delivered through the TD target, because Q^{t+1}_{tot} is conditioned on $\mathcal{C}^{t+1}$ and hence on G^{t+1}. A partner set that improves the team's subsequent value raises y^t and, through Equation 47, increases $Q^t_i(a^t_i)$ for the actions that produced it. Relational attention additionally receives gradient within round t along a differentiable path, since the aggregated feature ξ^t_i of Equation 5 enters Q^t_i directly.*

C.3 Monotonicity and Individual–Global–Max Consistency

Let each eligible agent have a finite, nonempty action space $\mathcal{A}^t_i \subseteq \mathcal{A}_{\mathrm{comm}}$ and write

$$\mathcal{A}^t = \prod_{i \in \mathcal{I}^t_{\mathrm{comm}}} \mathcal{A}^t_i. \tag{49}$$

The context and the mixing coefficients are held fixed while the elements of $\mathcal{A}^t$ are compared. In particular a^{off} and a^{solo} are ordinary candidates: their effect is realized in $\mathcal{I}^{t+1}_{\mathrm{comm}}$ and G^{t+1} through Equation 30, not in the current attention normalization. By Equation 45,

$$\frac{\partial Q^t_{\mathrm{tot}}}{\partial Q^t_i(a_i)} = \widehat{\beta}^t_i \;\geq\; 0 \qquad \forall i \in \mathcal{I}^t_{\mathrm{comm}}, \tag{50}$$

so the mixer is monotonic in every individual utility. Define

$$\mathcal{M}^t_i = \operatorname*{arg\,max}_{a_i \in \mathcal{A}^t_i} Q^t_i(a_i), \qquad \mathcal{P}^t = \left\{i : \widehat{\beta}^t_i > 0\right\}, \qquad \mathcal{Z}^t = \left\{i : \widehat{\beta}^t_i = 0\right\}. \tag{51}$$

Theorem C.8 (IGM consistency)**.** *For the fixed context $\mathcal{C}^t$,*

$$\operatorname*{arg\,max}_{\boldsymbol{a} \in \mathcal{A}^t} Q^t_{\mathrm{tot}}(\mathcal{C}^t, \boldsymbol{a}) = \left(\prod_{i \in \mathcal{P}^t} \mathcal{M}^t_i\right) \times \left(\prod_{i \in \mathcal{Z}^t} \mathcal{A}^t_i\right), \tag{52}$$

so every independently greedy joint action is a joint maximizer. If in addition the attention logits are finite and

$$W^t := \sum_{h=1}^{H} w^t_h \;>\; 0, \tag{53}$$

then $\widehat{\beta}_i^t > 0$ *for every* $i \in \mathcal{I}_{\mathrm{comm}}^t$ *and the set-valued IGM identity holds exactly:*

$$\arg\max_{\boldsymbol{a}\in\mathcal{A}^t} Q_{\mathrm{tot}}^t(\mathcal{C}^t, \boldsymbol{a}) = \prod_{i\in\mathcal{I}_{\mathrm{comm}}^t} \arg\max_{a_i\in\mathcal{A}_i^t} Q_i(\xi_i^t, a_i). \tag{54}$$

Proof. Let $m_i^t = \max_{a_i\in\mathcal{A}_i^t} Q_i^t(a_i)$, which exists because $\mathcal{A}_i^t$ is finite and nonempty. Since $\widehat{\beta}_i^t \geq 0$ by Equation 45, each summand of Equation 44 obeys $\widehat{\beta}_i^t Q_i^t(a_i) \leq \widehat{\beta}_i^t m_i^t$, hence

$$Q_{\mathrm{tot}}^t(\mathcal{C}^t, \boldsymbol{a}) \;\leq\; b^t + \sum_{i\in\mathcal{I}_{\mathrm{comm}}^t} \widehat{\beta}_i^t m_i^t \tag{55}$$

for every $\boldsymbol{a} \in \mathcal{A}^t$, and the bound is attained. Equality in Equation 55 holds if and only if $Q_i^t(a_i) = m_i^t$, that is $a_i \in \mathcal{M}_i^t$, for every $i \in \mathcal{P}^t$, while for $i \in \mathcal{Z}^t$ the term $\widehat{\beta}_i^t Q_i^t(a_i)$ vanishes and the choice of a_i leaves Q_{tot}^t unchanged. This establishes Equation 52, and $\mathcal{M}_i^t \subseteq \mathcal{A}_i^t$ gives the subsequent claim. For the second part, finite logits make the softmax of Equation 43 strictly positive, so $\lambda_{i,h}^t > 0$ for all i and h. If $W^t > 0$ there exists a head $h^\star$ with $w_{h^\star}^t > 0$, whence $\widehat{\beta}_i^t \geq w_{h^\star}^t \lambda_{i,h^\star}^t > 0$ for every eligible agent. Therefore $\mathcal{Z}^t = \varnothing$ and Equation 52 reduces to Equation 54. □

If all head weights vanish, then $Q_{\mathrm{tot}}^t = b^t$ for every candidate joint action; independently greedy actions remain joint maximizers by Equation 52, but the two maximizing sets need not coincide. The condition $W^t > 0$ of Equation 53 can be enforced through a strictly positive head-weight parameterization whenever the set-valued identity is required.

Remark C.9 (Reachable coefficients and the cost of monotonicity)**.** *Because* $\lambda_{\cdot,h}^t$ *is normalized over* $\mathcal{I}_{\mathrm{comm}}^t$*, the learned coefficients satisfy*

$$\sum_{i\in\mathcal{I}_{\mathrm{comm}}^t} \widehat{\beta}_i^t = \sum_{h=1}^{H} w_h^t \sum_{i\in\mathcal{I}_{\mathrm{comm}}^t} \lambda_{i,h}^t = W^t. \tag{56}$$

The mixer therefore redistributes a single state-dependent mass W^t *across agents rather than rescaling each utility independently, and by Equation 50 it cannot represent the negative sensitivities that Equation 41 permits. Both restrictions are the price of monotonicity, which is what makes Equation 54 and hence decentralized greedy execution available (Rashid et al., 2018). What* $\widehat{\beta}_i^t$ *adds relative to a state-only mixer is not expressiveness in the utilities but the dependence of the normalized credit profile on* $\mathcal{N}_i^{\mathrm{in},t}$ *and* $\mathcal{N}_i^{\mathrm{out},t}$ *established in Corollary C.4; the empirical claim of this paper is that this monotonic, topology-conditioned class suffices for the communication action space of Section 3.2, not that it is universal.*

Remark C.10 (Scope of the consistency result)**.** *Theorem C.8 relies on the temporal ordering of Eqs. equation 29–equation 30: all candidate actions at round* t *share one pre-decision context and therefore one set of mixing coefficients, and the selected action determines only the next topology. Without this ordering—for instance if the mixer were conditioned on the topology induced by the candidate action itself—the coefficients would vary across* $\mathcal{A}^t$*, the decomposition in Equation 44 would no longer be a fixed weighted sum, and monotonicity in each* Q_i^t *would not imply Equation 54.*

D ACTS TRAINING AND DECENTRALIZED EXECUTION

Algorithm 1 ACTS Centralized Training

Require: Training tasks $\mathcal{D}_{\text{train}}$; roles $\{\rho_i\}_{i=1}^{N}$; episodes M; horizon T; discount factor γ; objective weights $w_{\text{task}}, w_{\text{tok}}$; token scale B; initial exploration rate ϵ_0; minibatch size b; target update interval K
Ensure: Online parameters Θ and target parameters Θ^-

Initialize Θ, $\Theta^- \leftarrow \Theta$, replay buffer $\mathcal{B} \leftarrow \varnothing$, $\epsilon \leftarrow \epsilon_0$, and update counter $u \leftarrow 0$
for $e = 1, \dots, M$ **do**
  Sample $x \sim \mathcal{D}_{\text{train}}$; initialize G^1, $\mathcal{I}_{\text{comm}}^1 = \mathcal{I}$, $h_i^0 = \mathbf{0}$, trajectory $\tau \leftarrow ()$, and $C_{\text{tok}} \leftarrow 0$
  for $t = 1, \dots, T$ **do**
    Encode the pre-decision context $\mathcal{C}^t$ using Equations 3–9
    Select $\boldsymbol{a}^t$ with ϵ-greedy local utilities; assign $a_i^t = a^{\text{off}}$ to inactive agents
    Construct $\mathcal{I}_{\text{comm}}^{t+1}$ and G^{t+1} using Equations 1 and 7
    Set $d^t \leftarrow \mathbf{1}[|\mathcal{I}_{\text{comm}}^{t+1}| < \lceil N/2 \rceil \ \vee \ t = T]$
    Execute round t and update C_{tok}; set $r^t \leftarrow 0$
    if $d^t = 0$ **then**
      Observe s^{t+1} and $\{o_i^{t+1}\}$
    else
      Set terminal next-state fields to $\perp$
    end if
    Append $\zeta^t = (s^t, \boldsymbol{o}^t, G^t, \mathcal{I}_{\text{comm}}^t, \boldsymbol{a}^t, r^t, s^{t+1}, \boldsymbol{o}^{t+1}, G^{t+1}, \mathcal{I}_{\text{comm}}^{t+1}, d^t)$ to τ
    if $d^t = 1$ **then**
      break
    end if
  end for
  Aggregate the final answer and compute $R \leftarrow w_{\text{task}} R_{\text{task}} - w_{\text{tok}} \min(C_{\text{tok}}/B, 1)$
  Set the reward of the final transition in τ to R and add all transitions in τ to $\mathcal{B}$
  if $|\mathcal{B}| \geq b$ **then**
    Sample a minibatch $\{\zeta_k\}_{k=1}^{b}$ from $\mathcal{B}$
    Compute

$$y_k = \begin{cases} r_k, & d_k = 1, \\ r_k + \gamma Q_{\text{tot}}^-(\boldsymbol{a}_k^{*,+}; \mathcal{C}_k^+), & d_k = 0, \end{cases}$$

    where $\boldsymbol{a}_k^{*,+}$ is selected using the target local utilities
    Update Θ by minimizing

$$\mathcal{L}_{\text{TD}} = \frac{1}{2b} \sum_{k=1}^{b} [y_k - Q_{\text{tot}}(\boldsymbol{a}_k; \mathcal{C}_k)]^2$$

    with y_k detached and the backbone LLM frozen
    Set $u \leftarrow u + 1$
    if $u \bmod K = 0$ **then**
      $\Theta^- \leftarrow \Theta$
    end if
  end if
  Anneal ϵ
end for

We reuse the GNN–GRU encoder and relational-attention/Q branch of Section 3.2 and the topology encoder and topological mixer of Section 3.3. For $G^t = (\mathcal{I}, \mathcal{E}^t)$, write $\mathcal{N}_i^{\text{in},t} = \{j \mid (j,i) \in \mathcal{E}^t\}$ and $\mathcal{N}_i^{\text{out},t} = \{j \mid (i,j) \in \mathcal{E}^t\}$ for the directed neighborhoods encoded by z_i^t in equation 9. The operator BUILDTOPOLOGY denotes the topology construction map defined in Appendix B. The six actions are listed in Appendix E.3, and the observation and state vectors are specified in Appendix E.2. The action a^{off} suppresses both backbone inference and communication. The action a^{solo} invokes one independent backbone inference without creating communication edges, and its output is retained for final answer aggregation. The corresponding agent is excluded from subsequent communication rounds. All other actions produce an output and induce communication edges according to the selected mode. These messages enter the observations of round $t+1$.

Every round is organized around the pre-decision context $\mathcal{C}^t = (s^t, G^t, \mathcal{I}_{\text{comm}}^t, \{\xi_i^t, z_i^t\}_{i \in \mathcal{I}_{\text{comm}}^t})$, computed from the realized topology before any action is selected and held fixed while candidate

actions are compared, so the coefficients $\widehat{\beta}_i^t = \sum_h w_h^t \lambda_{i,h}^t$ are constants of $\mathcal{C}^t$ and equation 12 applies. Since agents outside $\mathcal{I}_{\text{comm}}^t$ are assigned a^{off}, equation 1 gives $\mathcal{I}_{\text{comm}}^{t+1} \subseteq \mathcal{I}_{\text{comm}}^t$. An agent selecting a^{off} or a^{solo} therefore withdraws from subsequent communication, although only a^{off} suppresses the current-round backbone inference. Hard partner-set construction is nondifferentiable, so relational attention is supervised only through the soft feature ξ_i^t while topological attention is trained through the mixer; the target branch is detached, and replay stores raw inputs and topologies rather than cached hidden states.

Algorithm 2 ACTS Decentralized Execution

Require: Query x; roles $\{\rho_i\}_{i=1}^N$; horizon T; trained parameters Θ
Ensure: Team answer Y

$G^1 \leftarrow$ complete digraph on $\mathcal{I}$ without self-loops; $\mathcal{I}_{\text{comm}}^1 \leftarrow \mathcal{I}$; $h_i^0 \leftarrow \mathbf{0}$
for $t = 1, \dots, T$ **do**
  Build o_i^t from x, agent i's history, and the messages delivered under G^t
  For $i \in \mathcal{I}_{\text{comm}}^t$: h_i^t, ξ_i^t, $\{\alpha_{ij}^t\}$, then $a_i^t \leftarrow \arg\max_{a \in \mathcal{A}_{\text{comm}}} Q_i(\xi_i^t, a)$ ▷ Equation 3–equation 5, equation 12
  $a_i^t \leftarrow a^{\text{off}}$ for $i \notin \mathcal{I}_{\text{comm}}^t$; form $\boldsymbol{a}^t$ and $\boldsymbol{\alpha}^t$
  $\mathcal{I}_{\text{comm}}^{t+1} \leftarrow \{i \mid a_i^t \notin \{a^{\text{off}}, a^{\text{solo}}\}\}$; $G^{t+1} \leftarrow$ BUILDTOPOLOGY$(\boldsymbol{a}^t, \boldsymbol{\alpha}^t)$ ▷ Equation 1, equation 7
  Run round t (a^{off}: no call; a^{solo}: own answer; else message along G^{t+1}) and append the outputs to the local histories
  if $|\mathcal{I}_{\text{comm}}^{t+1}| < \lceil N/2 \rceil$ **then break**
end for
return Y from the answer-aggregation node

E Experimental Details

E.1 Training and Evaluation Details

Benchmarks and backbone models. We evaluate ACTS on nine benchmarks across four domains. The coding benchmarks are LiveCodeBench v6 and HumanEval (Jain et al., 2024; Chen et al., 2021). The reasoning benchmarks are MMLU-Pro and the first 250 questions of Humanity's Last Exam (Wang et al., 2024; Phan et al., 2025). HotpotQA is used for multi-hop question answering (Yang et al., 2018). The mathematical benchmarks are AIME 2025, AIME 2026, HMMT 2025, and Beyond-AIME (Mathematical Association of America, 2025; 2026; MathArena, 2025; ByteDance-Seed, 2025). Each AIME benchmark comprises 30 problems from the corresponding AIME I and AIME II examinations. Experiments use GPT-OSS-120B and DeepSeek-V4-Flash as frozen backbone LLMs. Experiments use GPT-OSS-120B and DeepSeek-V4-Flash as frozen backbone LLMs. Each task is instantiated with three role-conditioned agents followed by a final answer-aggregation node.

Baselines and controlled comparison. We compare ACTS with direct prompting, LLM-Debate (Du et al., 2023), AutoGen (Wu et al., 2023), LangGraph (LangChain AI, 2024), Microsoft Agent Framework (Microsoft, 2025), TopoDIM (Sun et al., 2026), and Agent Q-Mix (Jiang et al., 2026). All methods use the same team size ($N = 3$), task prompts, communication-round budget, and answer-evaluation procedure.

Training protocol and evaluation. The ACTS scheduler is trained for 50 episodes using 15 training examples per domain. The communication budget is three rounds for mathematical tasks and two rounds for coding and reasoning tasks. All trainable modules are optimized with Adam using learning rate 5×10^{-4} and discount factor $\gamma = 0.99$. The replay buffer has capacity $5{,}000$, the mini-batch size is 8, and ϵ-greedy exploration is annealed linearly from 1.0 to 0.05 over 50 episodes. The learned parameters are fixed during evaluation. Reported results are averaged over three independent runs with different random seeds.

E.1.1 Data Sampling and Information Gain Computation

For each model M, benchmark D, and topology family C, empirical communication gain is defined as $\Delta(M, D, C) = \text{mean}_{m \in C} \text{ACC}(m, M, D) - \text{ACC}(\text{Direct}, M, D)$, with accuracy expressed as a fraction. The threshold $x = 2/n$ is a heuristic tolerance corresponding to two correct answers

Table 3: Per-benchmark evaluation samples, IG thresholds, and classifications under GPT-OSS-120B and DeepSeek-V4-Flash.

Benchmark	Domain	n	$x = 2/n$	IG (GPT-OSS-120B)	IG (DeepSeek-V4-Flash)
LiveCodeBench	Code	175	0.0114	> 0	> 0
HumanEval	Code	164	0.0122	≈ 0	≈ 0
HLE	Reasoning	250	0.0080	≈ 0	≈ 0
HotpotQA	Multi-hop QA	740	0.0027	> 0	> 0
MMLU-Pro	Reasoning	70	0.0286	> 0	> 0
AIME 2025	Math	30	0.0667	> 0	> 0
AIME 2026	Math	30	0.0667	> 0	> 0
Beyond-AIME	Math	100	0.0200	> 0	> 0
HMMT 2025	Math	30	0.0667	> 0	≈ 0

among n evaluation samples. A positive gain is identified when at least one topology family satisfies $\Delta(M, D, C) > x$. Otherwise, no positive gain is identified under this criterion. This descriptive threshold does not establish statistical significance.

E.2 States and Observations

ACTS follows a centralised-training, decentralised-execution scheme. A shared per-agent value network receives a local observation o_i for each agent i, while a mixing network integrates the agent-specific values into a joint estimate Q_{tot} conditioned on a global state s. We describe the construction of each.

Local observation. For $N = 3$, each agent receives a 51-dimensional observation o_i^t comprising a 32-dimensional task embedding, a 16-dimensional one-hot agent identifier, a binary indicator of whether the agent has produced an answer, its incident edge count, and the normalized cumulative team token consumption clipped to $[0, 1]$.

Global state. The global state s^t concatenates all agents' local observations, three topology statistics comprising the edge count, graph density, and agent count, and the flattened $N \times N$ relational-attention matrix. For $N = 3$, the resulting state dimension is $3 \times 51 + 3 + 9 = 165$.

E.3 Action Space

At each round every active agent selects one of six discrete actions, grouped into three roles. Three actions, *off*, *solo*, and *broadcast*, operate outside the attention matrix: off deactivates the agent entirely for the round, emitting no edges and issuing no LLM call, and once a majority of agents are inactive the episode terminates early; solo produces an independent answer and emits no edges; broadcast sends the agent's current output to every other agent. Two actions build sparse directed edges from the learned attention matrix a_{edge}: *attn-broadcast* opens an edge $i \to j$ when the attention that agent j pays to i exceeds the row mean, and *attn-receive* requests information only from the neighbours that agent i itself attends to. The remaining action, *receive*, performs a generic information fetch. Figure 7 reports the learned distribution of these six actions across the nine benchmarks.

E.4 Case Study

Figure 8 illustrates the same machinery at the trajectory level on one HotpotQA question and one HLE question. On HotpotQA the agents remain active throughout: several agents take the broadcast action, sending their analysis of the question to the team while one agent retains the solo action as an independent reference, after which the broadcasting agents switch to attn-receive, so that the initially connected topology contracts into a sparse pull while the informative edges are preserved. On HLE one agent settles the answer under the solo action, a single attn-broadcast check is routed only to the agent that requires it, and the remaining agents are successively deactivated through the off action, with the majority-inactive condition terminating the episode early. The two trajectories correspond to the two mechanisms that the aggregate action shares in Figure 6 average over: selective routing when the agents contribute complementary information, and deactivation when a single perspective suffices.

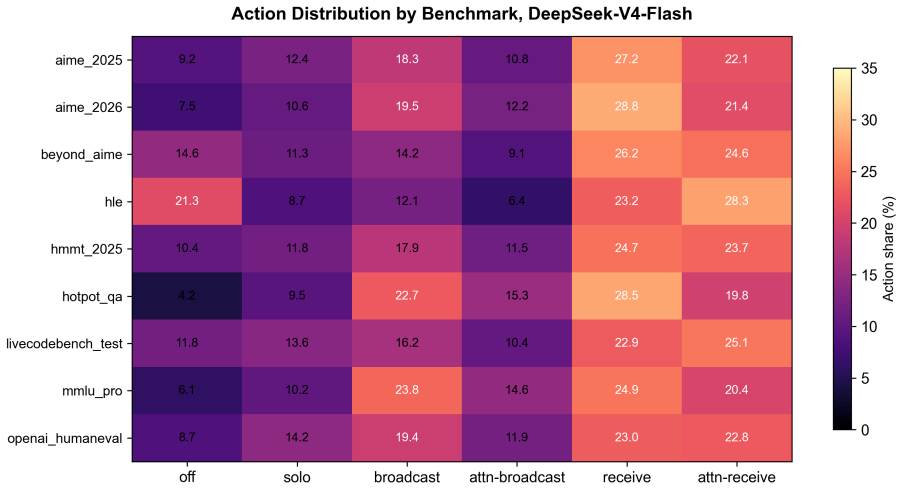

Figure 7: Distribution of the six actions across benchmarks under DeepSeek-V4-Flash. Each cell gives the proportion of rounds in which the corresponding action is selected. The learned policy favours broadcast and attn-broadcast on benchmarks where the agents contribute complementary information, and shifts toward off and attn-receive where a single perspective suffices.

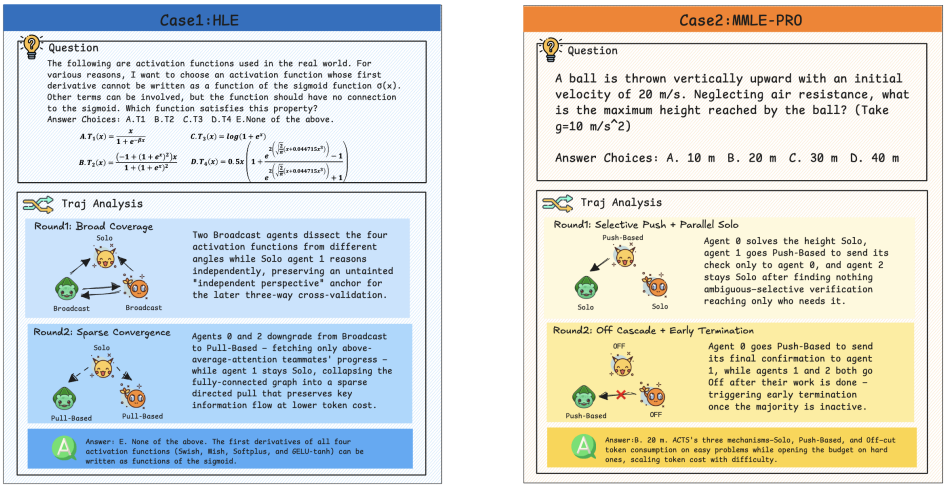

Figure 8: Trajectory-level case study on one HotpotQA question and one HLE question. On HotpotQA the connected topology contracts into a sparse pull while one agent retains the solo action as an independent reference. On HLE one agent settles the answer under the solo action, a single attn-broadcast check is routed to the agent that requires it, and the remaining agents are deactivated through the off action, triggering early termination under the majority-inactive condition.

Table 4: Average per-sample token usage ($\times 10^3$) on HotpotQA, a high-information-gain (IG > 0) task, and HLE, a low-information-gain (IG ≈ 0) task, for both backbone models. ACTS (highlighted) consumes the fewest tokens among all multi-agent methods in every column; only single-agent prompting uses fewer, at a large accuracy cost.

	HotpotQA		HLE	
Method	GPT-OSS-120B	DeepSeek-V4-Flash	GPT-OSS-120B	DeepSeek-V4-Flash
Direct Prompting	5.8	3.6	4.5	2.8
LLM-Debate	14.1	9.5	12.3	8.2
AutoGen	16.0	10.8	13.8	9.3
LangGraph	16.4	11.0	14.0	9.5
Agent Framework	16.7	11.3	14.2	9.7
G-Designer	12.5	8.4	10.8	7.2
TopoDIM	11.4	7.6	9.8	6.6
GTD	12.1	8.1	10.5	7.0
MaAS	11.8	7.9	10.2	6.8
Agent Q-Mix	8.7	5.8	7.9	5.3
ACTS	7.1	4.8	6.4	4.3

E.5 Per-Sample Token Usage

Table 4 lists the average per-sample token consumption of each method on HotpotQA, a high-information-gain (IG > 0) task, and HLE, a low-information-gain (IG ≈ 0) task, for both backbone models. These are the numbers plotted against accuracy in Figure 4. Single-agent prompting uses the fewest tokens in every column, but at the cost of substantially lower accuracy; ACTS consumes the fewest tokens among all multi-agent methods in every column, while the static topologies are the most expensive because they broadcast the full running transcript to every agent each round.

E.6 Agent Role Assignments and Prompts

Each agent in the multi-agent system is assigned one of six predefined roles. Roles are allocated in round-robin fashion by agent index modulo 6. Since our experiments use $n = 3$ agents, only the first three roles are instantiated in any given run:

R1: Independent Solver — solves the problem autonomously without relying on peer input.

R2: Team Solver — approaches the problem collaboratively, integrating suggestions from teammates.

R3: Questioner — probes assumptions and raises critical questions about the problem.

The remaining three roles (Refiner, Verifier, Debater) are defined in the role pool for future extensibility to larger agent teams, or for experiments where $n > 3$ (e.g., $n = 6$). The round-robin allocation modulo 6 naturally activates them when the team size increases, without any change to the prompt construction logic.

E.6.1 Agent System Prompt

Every agent receives a system prompt that instantiates its role within an n-agent team and instructs it on the expected output format. The general template is:

```
You are {ROLE} on a {n}-agent team.  {SUFFIX}
```

where {ROLE} is one of the six role labels listed above and {SUFFIX} is a question-type-specific instruction detailed below.

E.6.2 Question-Type Suffixes

The suffix appended to each agent's system prompt depends on the `answer_type` of the question. Five answer types are supported:

numerical. *"Solve the problem step by step, then state the final numerical answer. End with a line 'Final Answer: <number>'."*

multiple_choice. *"This is a single-choice question; there is exactly one correct option. The options may be labeled with any letter from A up to Z (not just A–E), so read all options carefully before choosing. Briefly analyze why each candidate is right or wrong, then commit to exactly one option. End with a line 'Final Answer: <letter>' (e.g., 'Final Answer: G')."*

text. *"Answer concisely. If a number, give the number. If a name, give the name. End with a line 'Final Answer: <answer>'."*

open_answer. *"Answer concisely in a few words. End with a line 'Final Answer: <answer>'."*

code. *"Write the complete, runnable solution code. Output ONLY the code—no markdown fences, no explanation."*

E.6.3 User Message

The user message delivered to each agent is constructed by concatenating the following components in order:

1. The question text.
2. Answer options (only for `multiple_choice` questions), formatted as `A. ...`, `B. ...`, etc.
3. Reference passages, when provided by the dataset (e.g., supporting documents for multi-hop QA).
4. Previously generated answers from inbound neighbors in the communication topology, listed as `Agent {i}: {answer}`.

E.6.4 Final Decision Maker

After all agents have executed across all rounds, a **Final Decision Maker** reviews the collected answers and produces the single final output. Its system prompt is:

```
You are the Final Decision Maker of a {n}-agent team. Read
the task and all agents' answers carefully, then give the
correct single answer.
```

The corresponding user message includes the original question, options (if applicable), reference context, and the full list of agent answers, followed by the instruction `Give the final answer.`

E.6.5 No-Role Baseline

For ablation experiments, we also evaluate a variant where all agents share an identical, role-agnostic system prompt:

```
You are a general team member on a {n}-agent team. {SUFFIX}
```

This removes the role specialization signal while keeping all other aspects of the prompt (question-type suffix, user message structure, and communication topology) unchanged.