# Superficial Success vs. Internal Breakdown: An Empirical Study of Generalization in Adaptive Multi-Agent Systems

**Namyoung So**[†] **Seokgyu Jang**[†] **Taeuk Kim**[*]
Department of Computer Science, Hanyang University, Seoul, Republic of Korea
{thskadud, diamondgyu, kimtaeuk}@hanyang.ac.kr

## Abstract

Adaptive multi-agent systems (MAS) are increasingly adopted to tackle complex problems. However, the narrow task coverage of their optimization raises the question of whether they can function as general-purpose systems. To address this gap, we conduct an extensive empirical study of adaptive MAS, revealing two key findings: (1) **topological overfitting**—they fail to generalize across different domains; and (2) **illusory coordination**—they achieve reasonable surface-level accuracy while the underlying agent interactions diverge from ideal MAS behavior, raising concerns about their practical utility. These findings highlight the pressing need to prioritize generalization in MAS development and motivate evaluation protocols that extend beyond simple final-answer correctness.

## 1 Introduction

Alongside recent advances in large language models (LLMs) (Google, 2025; OpenAI, 2025), multi-agent systems (MAS) (Qian et al., 2024; Khan et al., 2024; Chen et al., 2024; Yu et al., 2024)—which treat each model as an agent and coordinate multiple agents to accomplish challenging tasks—have attracted growing attention. In this paradigm, agents collaborate as a unified system, iteratively exchanging feedback to refine their conclusions.

Among various lines of work, **adaptive MAS** (Zhuge et al., 2024; Zhang et al., 2025; Li et al., 2025) have emerged as a prominent direction, where agent roles and communication topologies are adapted to a given task and objective, analogous to optimization in standard supervised learning.[1]

Yet this trend is paradoxical: although adaptive MAS are built from general-purpose LLMs, they are often heavily tailored to a narrow set of tasks.

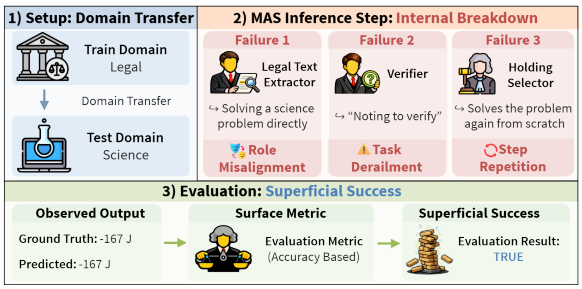



Figure 1: Example of superficial success vs. internal breakdown under domain transfer. When an adaptive MAS trained on the legal domain is applied to science, the agents make multiple errors during collaboration, yet the final answer remains correct due to the strength of individual LLMs—illustrating illusory coordination.

Consequently, despite their potential for broad generalization, much of the literature focuses on optimizing MAS for specific domains, leaving it unclear whether such systems generalize beyond their training scope. This is not merely a theoretical concern: constructing a MAS entails substantial costs from orchestrating LLMs, making it impractical to deploy a separate MAS for each task of interest.

In this work, we investigate the generalization capabilities of adaptive MAS approaches and identify two failure modes. First, we show that **topological overfitting** (§3)—performance degradation when an adaptive MAS is evaluated on out-of-distribution (OOD) tasks—is prevalent, revealing a clear failure of domain and task transfer.

Meanwhile, even MAS that appear to generalize across domains often do so for the wrong reasons. As illustrated in Figure 1, closer inspection of their execution traces reveals that collaboration mechanisms frequently break down, and the system instead relies on the brute-force reasoning of individual LLMs rather than collective intelligence—a phenomenon we term **illusory coordination** (§4).

To better understand this issue, we conduct complementary qualitative and quantitative analyses.

[†]Equal contribution. [*]Corresponding author.

[1]We use the term *topology* to denote the configurations of both agent roles (nodes) and their connections (edges).



*Findings of the Association for Computational Linguistics: ACL 2026*, pages 15328–15354
July 2-7, 2026 

| Training / Test (Domain) | L | D | MH | S | MA | CS |
|---|---|---|---|---|---|---|
| CaseHOLD (**L**egal) | **63.5** | 44.2 | 57.4 | 41.8 | 65.5 | 70.0 |
| COM$^2$ (**D**etective) | 53.4 | **47.9** | 53.8 | 35.8 | 54.4 | 19.5 |
| MuSiQue (**M**ulti-**H**op) | 63.2 | 49.0 | **58.4** | 40.1 | 65.4 | 73.8 |
| SciBench (**S**cience) | 61.8 | 34.2 | 54.9 | **38.9** | 62.8 | 47.5 |
| TheoremQA (**M**ath) | 62.2 | 47.2 | 57.5 | 36.9 | **63.8** | 75.1 |
| StrategyQA (**C**ommonsense) | 0.6 | 0.5 | 41.5 | 0.1 | 15.7 | **72.5** |
| Multi-Domain Training | 60.2 | 46.7 | 52.9 | 41.1 | 64.4 | 75.3 |

(a) AgentDropout

| Training / Test (Domain) | L | D | MH | S | MA | CS |
|---|---|---|---|---|---|---|
| CaseHOLD (**L**egal) | **61.7** | 41.2 | 47.6 | 40.8 | 66.4 | 68.9 |
| COM$^2$ (**D**etective) | 63.3 | **40.8** | 49.6 | 38.9 | 62.3 | 55.2 |
| MuSiQue (**M**ulti-**H**op) | 51.7 | 41.4 | **48.1** | 39.3 | 62.0 | 48.9 |
| SciBench (**S**cience) | 56.0 | 39.2 | 47.6 | **33.0** | 57.2 | 71.5 |
| TheoremQA (**M**ath) | 48.5 | 37.8 | 46.8 | 35.9 | **59.8** | 58.1 |
| StrategyQA (**C**ommonsense) | 38.2 | 34.4 | 33.8 | 34.2 | 56.0 | **73.8** |
| Multi-Domain Training | 62.2 | 40.6 | 52.1 | 39.0 | 65.9 | 73.5 |

(b) AFlow

Table 1: In-domain and OOD performance of AgentDropout and AFlow on `GPT-oss-20B`. Cell colors are normalized per column by the column-wise maximum: values $\geq 95\%$ max are shaded in blue (i.e., successful transfer), and values $< 70\%$ max are in red (i.e., failed transfer). While domain transfer is often reasonable, it fails in many others. Multi-domain training shows promising results compared to **in-domain** (bolded, diagonal) baselines. See Appendix C for results on `Qwen3-30B-A3B` and full three-run details.

Qualitatively, we apply the Multi-Agent System Failure Taxonomy (MAST) of Cemri et al. (2025) and find that approximately 60% of failures in adaptive MAS stem from role non-adherence and miscommunication between agents.

Quantitatively, we propose **Role Alignment** ($\mathcal{R}$) and **Connection Significance** ($\mathcal{O}$)—two new metrics that respectively evaluate role preservation and inter-agent information flow—and use them to formalize illusory coordination: a regime where accuracy remains high despite $\mathcal{R}$ and/or $\mathcal{O}$ being low. Overall, topologies learned by adaptive MAS are often excessively tailored to specific tasks, giving rise to various forms of internal collapse.

In summary, our findings suggest that the community should look beyond final-answer accuracy and adopt more rigorous evaluation schemes that examine the internal dynamics of collaboration.

## 2 Background

**Formulation of Adaptive MAS** An *adaptive* MAS is an agent collaboration framework defined by a tuple $(\mathcal{A}, \mathcal{C})$, where $\mathcal{A}$ denotes the set of agents and $\mathcal{C}$ their connections.[2] Notably, both $\mathcal{A}$ and $\mathcal{C}$ are learned from data: given a task-specific training set $\mathcal{D}_{\text{train}}$ and a fixed base LLM, an optimization method $\mathbb{M}$ searches for an optimal topology:

$$(\mathcal{A}^*, \mathcal{C}^*) = \mathbb{M}(\mathcal{D}_{\text{train}}, \text{LLM}),$$

where $\mathcal{A}^*$ and $\mathcal{C}^*$ denote the optimized agent roles and connections, respectively.

As the base LLM, we employ `GPT-oss-20B` (Agarwal et al., 2025). Additional results with `Qwen3-30B-A3B` (Team, 2025), which show similar trends, are reported in Appendix C.

[2]Here, an agent is an LLM instantiated with a specific role and instructions; a single LLM may serve as multiple agents.

| Dataset | Domain | Specification |
|---|---|---|
| CaseHOLD (Chalkidis et al., 2022) | **L**egal | Judicial decision (holding) identification from case contexts. |
| COM$^2$ (Xiong et al., 2025) | **D**etective | Perpetrator identification under multi-logical constraints. |
| MuSiQue (Trivedi et al., 2022) | **M**ulti-**H**op | QA requiring integration across multiple long-context documents. |
| SciBench (Wang et al., 2023) | **S**cientific | University-level scientific reasoning and calculations. |
| TheoremQA (Chen et al., 2023) | **M**ath | Theorem-based STEM problem solving. |
| StrategyQA (Geva et al., 2021) | **C**ommonsense | Multi-step fact synthesis for boolean QA. |

Table 2: Overview of the six datasets used in our study, selected to cover diverse domains and reasoning types.

**Adaptive MAS Algorithms** We evaluate two representative algorithms: **AFlow** (Zhang et al., 2024c), which incrementally constructs communication paths (bottom-up), and **AgentDropout** (Wang et al., 2025), which prunes redundant links from a fully connected graph (top-down). AFlow jointly optimizes $\mathcal{A}$ and $\mathcal{C}$ during search, whereas AgentDropout optimizes only $\mathcal{C}$ after $\mathcal{A}$ has been determined. For AgentDropout, we use AgentInit (Tian et al., 2025) to optimize $\mathcal{A}$ beforehand.

For both ID and OOD evaluation, we use the same learned topology without further adaptation, varying only the test domain. Further procedural details are provided in Appendix B. To support reproducibility, we include the judging prompts (Appendix F), three-run results (Appendix C), and the full evaluation configurations (Appendix G).

**Datasets** We employ six datasets that span diverse domains and reasoning challenges (see Table 2). Adaptive MAS have primarily been developed with an emphasis on data-efficient optimization under limited training data (Zhang et al.,

| Failure Category | Example |
|---|---|
| • Disobey role specification | **Question:** Calculate the Carnot efficiency on given condition...<br>↪ **Legal Text Extractor**: 1. Convert the temperatures: ..., 2. Compute the efficiency: ... Final solution: 0.107 |
| • Step repetition<br>• No verification<br>• Ignores input | **Question:** Two sets of points are linearly separable when... true or false?<br>↪ **Clue Fact Extractor:** Two sets are linearly separable when... If the hulls are disjoint, ... The answer is: True<br>↪ **Validator:** If they are linearly separable, then... Consequently, ... The answer is: True |
| • Disobey task specification<br>• Task derailment | **Question:** What actions could have been taken to prevent crime when... (A):..., (B):..., ...<br>↪ **Context Analyzer:** ... (B) and (D) are good for purposes, while other candidates are...<br>↪ **Answer Synthesizer:** True |

Table 3: Case studies of internal breakdown. Reasoning traces flow downward (e.g., Clue Fact Extractor → Validator). Domain shifts induce multiple failure modes: role misalignment (the Legal Text Extractor attempts physics calculations), input neglect (the Validator disregards prior inputs and re-solves the problem), and task violation (the Answer Synthesizer responds "True" to a multiple-choice question), all leading to undesired outcomes.

2024a; Wang et al., 2025; Zhang et al., 2024c,b). To remain faithful to these original experimental settings, we cap the number of training samples at 100 for AFlow and 60 for AgentDropout.[3] Dataset statistics are reported in Appendix A.

## 3 Finding 1: Topological Overfitting

We first investigate the capacity of adaptive MAS topologies trained on specialized domains. All experiments are conducted with three separate runs.

### 3.1 Evaluation Criteria

**In-Domain (ID) Performance** Accuracy is measured on the test split of the source dataset used for optimization, following standard practice in the literature (Zhang et al., 2024c; Wang et al., 2025).

**Out-of-Distribution (OOD) Performance** To probe cross-domain robustness, we evaluate MAS topologies learned in one domain on datasets from unseen domains. Under our setup, OOD evaluation spans the five domains excluded from training.

**Multi-Domain Performance** As a simple recipe to mitigate overfitting, we test multi-domain training by mixing data from six domains while keeping the total number of training instances consistent.

### 3.2 Experimental Results

Table 1 shows that single-domain MAS optimization often leads to over-specialization that fails under distribution shifts. They exhibit a marked decline in OOD performance—e.g., on the legal domain, AgentDropout drops from 63.5% in-domain to 55.78% OOD (avg.). In addition, MAS optimized for commonsense reasoning often fail

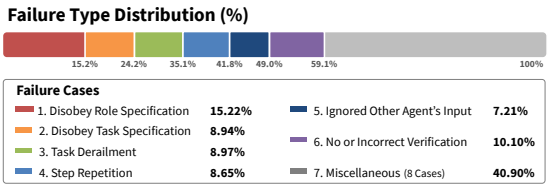



Figure 2: Failure types under domain transfer, grouped by MAST. Role-related and connection-related failures (Cases 1–6) account for around 60%, indicating they are primary sources of collaboration breakdown.

to transfer to numerical or multiple-choice problems. In such cases, accuracy degrades severely, as topologies optimized for binary (true/false) questions are unable to produce valid solutions in richer answer formats.

In contrast, a few cases show promising robustness—e.g., AgentDropout trained on MuSiQue and CaseHOLD—and the multi-domain variants also perform reasonably, suggesting potential for generalization. However, it remains unclear whether this apparent generalization reflects effective collaboration or simply the strength of individual agents.

## 4 Finding 2: Illusory Coordination

Motivated by the question posed in §3.2, we conduct an in-depth analysis of the inner workings of adaptive MAS from both qualitative and quantitative perspectives, focusing on AgentDropout.

### 4.1 Qualitative Analysis

As shown in Figure 1 and Appendix E, we discover cases where collaborations collapse under domain transfer. We organize these patterns using MAST (Cemri et al., 2025), which defines 14 MAS failure categories,[4] assigning 100 execution logs to one or more categories via an LLM judge (`GPT-oss-120B`).

[3] We compare the performance of AgentDropout with `GPT-oss-20B` using 60 vs. 200 training examples. The difference is negligible (Appendix C.3), supporting our decision to limit the instance count following prior conventions.

[4] Detailed definitions are provided in Appendix D.

| Training / Test (Domain) | L | D | MH | S | MA | CS |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | **1.00** | 0.56 | 0.04 | 0.22 | 0.25 | 0.54 |
| COM$^2$(**D**etective) | 0.79 | **1.00** | 0.04 | 0.43 | 0.47 | 0.82 |
| MuSiQue(**M**ulti-**H**op) | 0.69 | 1.00 | **0.38** | 0.50 | 0.58 | 0.58 |
| SciBench(**S**cience) | 0.44 | 0.49 | 0.04 | **1.00** | 0.62 | 0.46 |
| TheoremQA(**M**ath) | 0.38 | 0.36 | 0.04 | 0.60 | **1.00** | 0.32 |
| StrategyQA(**C**ommonsense) | 1.00 | 0.96 | 0.07 | 0.32 | 0.45 | **0.95** |
| Multi-Domain Training | 0.89 | 1.00 | 0.31 | 0.58 | 0.62 | 0.98 |

(a) Role Alignment ($\mathcal{R}$) Heatmap

| Training / Test (Domain) | L | D | MH | S | MA | CS |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | **1.00** | 0.07 | −1.79 | −2.07 | −1.89 | −1.56 |
| COM$^2$(**D**etective) | 0.90 | **1.00** | 0.17 | 0.65 | 0.58 | 0.46 |
| MuSiQue(**M**ulti-**H**op) | 1.00 | 0.96 | **0.15** | 0.95 | 0.80 | 0.86 |
| SciBench(**S**cience) | −0.75 | −0.50 | −0.57 | **1.00** | 0.77 | −0.07 |
| TheoremQA(**M**ath) | −0.12 | 0.21 | −0.07 | 1.00 | **0.88** | 0.48 |
| StrategyQA(**C**ommonsense) | 0.93 | 0.95 | 0.17 | 1.00 | 0.90 | **0.81** |
| Multi-Domain Training | 0.69 | 0.99 | −0.23 | 0.88 | 0.85 | 1.00 |

(b) Connection Significance ($\mathcal{O}$) Heatmap

Table 4: Illusory coordination of AgentDropout detected by $\mathcal{R}$ and $\mathcal{O}$. All entries are row-wise normalized by the maximum value in each row (i.e., each cell reports value / max(row)). Cell colors follow the normalized ratios: values $\geq 0.70$ are shaded blue (i.e., successful transfer), values $< 0.70$ are shaded red (i.e., failed transfer). In-domain results are in bold. Results provide new insights into MAS dynamics. Further details are in Appendix C.6.

In Figure 2, we focus on 6 topology-related failure types, including role violations and step repetition. They account for a majority (59.1%) of errors, suggesting that domain shift mainly induces systemic failures tied to role adherence and information flow. Table 3 further presents case studies of internal collapse in MAS.

## 4.2 Quantitative Analysis

Based on the qualitative analysis, we devise two novel metrics for additional quantitative analysis.

**Role Alignment** ($\mathcal{R}$) Our previous observations suggest that failures often stem from breakdowns in role adherence: agents in an MAS should preserve role diversity rather than collapse into identical behaviors. To quantify adherence to this principle, we introduce **Role Alignment** ($\mathcal{R}$).

Formally, let $\mathcal{A}$ denote the set of agents. For each agent $i \in \mathcal{A}$, let $p_i$ represent its role prompt and $y_i$ for its output. We utilize an encoder $\mathbf{e}(\cdot)$ (i.e., `all-MiniLM-L6-v2`) to map these texts into a shared embedding space. We then compute $S_{1,i}$, the cosine similarity between the agent's role definition and its output:

$$S_{1,i} = \cos\big(\mathbf{e}(p_i), \mathbf{e}(y_i)\big).$$

A higher $S_{1,i}$ suggests stronger semantic alignment with the assigned role. Second, we compute $S_{2,i}$, which represents the average similarity between agent $i$'s output and the outputs of all other agents $j \in \mathcal{A} \setminus \{i\}$ for the same instance:

$$S_{2,i} = \tfrac{1}{|\mathcal{A}|-1} \textstyle\sum_{j \neq i} \cos\big(\mathbf{e}(y_i), \mathbf{e}(y_j)\big).$$

A large $S_{2,i}$ indicates that agents produce generic, similar responses. Finally, Role Alignment for agent $i$ is defined as $\mathcal{R}_i = S_{1,i} \times (1 - S_{2,i})$. Thus, a large $\mathcal{R}_i$ signifies a robust topology where agents contribute unique, role-specific information.

| $\mathcal{O}_i$ | $\alpha_{i,\ell}$ | $s_{i,\ell}$ | **Interpretation** |
|---|---|---|---|
| $\approx 0$ | small or mixed | either | Negligible net message impact. |
| $< 0$ | large | $-1$ | Influential but unhelpful messages. |
| $> 0$ | large | $+1$ | Influential and useful messages. |

Table 5: Interpretation of $\mathcal{O}_i$ in terms of message influence $\alpha_{i,\ell}$ and usefulness $s_{i,\ell}$.

**Connection Significance** ($\mathcal{O}$) To diagnose information flow, such as identifying disregarded messages, we define **Connection Significance** ($\mathcal{O}$), which quantifies how incoming messages from other agents influence an agent's output $y_i$ beyond the contribution of the task query $q$ and the agent's role $p_i$. Here, a *message* refers to the output of a predecessor agent that serves as input to agent $i$.

Let $\mathcal{X}_i = \{x_1, \ldots, x_m\}$ denote the set of all incoming messages to agent $i$. We then compute influence weights $\alpha_{i,\ell}$ by comparing message utility against the static priors $p_i$ and $q$:

$$\alpha_{i,\ell} = \frac{\exp\big(sim(x_\ell, y_i)\big)}{\sum_{\hat{z} \in \mathcal{X}_i \cup \{p_i, q\}} \exp\big(sim(\hat{z}, y_i)\big)},$$

where $\hat{z}$ ranges over all prompt components in $\mathcal{X}_i \cup \{p_i, q\}$, and $sim(a, b) = \cos\big(\mathbf{e}(a), \mathbf{e}(b)\big)$. That is, $\alpha_{i,\ell}$ is obtained by applying a softmax over both the incoming messages and the static priors $(p_i, q)$. As a result, a message receives a smaller weight when $y_i$ is better explained by the agent's role $p_i$ or the task query $q$ than by the incoming messages.

Next, we determine the message usefulness $s_{i,\ell} = \text{LaaJ}(q, p_i, x_\ell) \in \{+1, -1\}$ using an LLM-as-a-judge (`GPT-oss-20B`),[5] and aggregate the weighted scores over all input messages:

$$\mathcal{O}_i = \textstyle\sum_{x_\ell \in \mathcal{X}_i} \alpha_{i,\ell}\, s_{i,\ell}.$$

[5]The LaaJ prompt is provided in Appendix F.

Thus, $\mathcal{O}_i$ captures the aggregate effect of all inputs to agent $i$, jointly reflecting their influence ($\alpha$) and usefulness ($s$): values near zero indicate negligible net impact, negative values indicate influential but unhelpful messages, and positive values reflect influential and useful ones (see Table 5).

**Formal Definition of Illusory Coordination** Finally, for each (train, test) pair, we report in Table 4 a single score for each metric, obtained by averaging $\mathcal{R}_i$ and $\mathcal{O}_i$ over all agents $i \in \mathcal{A}$ and then over test instances $\mathcal{D}_{\text{test}}$:

$$(\mathcal{R}, \mathcal{O}) = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{q \in \mathcal{D}_{\text{test}}} \left( \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} (\mathcal{R}_i(q), \mathcal{O}_i(q)) \right).$$

Building on these values, we define **illusory coordination** as settings where task accuracy is high while internal collaboration quality is poor—i.e., low $\mathcal{R}$ and/or low (possibly negative) $\mathcal{O}$. This definition is diagnostic: rather than claiming collaboration is always illusory, our goal is to identify cases where surface-level success is not supported by the intended collaborative mechanism.

**Results with $\mathcal{R}$** While Table 1 suggests that CaseHOLD (**L**) transfers well to SciBench (**S**), Table 4 reveals a different picture: $\mathcal{R}$ drops to only 22% of its in-domain value, indicating that this apparent success is not supported by role adherence. Conversely, when transferring from COM$^2$ (**D**) to StrategyQA (**CS**), $\mathcal{R}$ remains high (82% of the in-domain value) despite the poor accuracy reported in Table 1. These observations point to a *dissociation* between accuracy and role alignment. Table 6 further supports this interpretation: the Pearson correlation between accuracy and $\mathcal{R}$ (**Acc–$\mathcal{R}$**) is near zero in most cases, suggesting that final-answer correctness does not necessarily reflect whether the intended role structure is preserved.

**Results with $\mathcal{O}$** Table 4 reveals degradation in some OOD scenarios, where agents fail to integrate incoming messages and instead rely on independent reasoning—a hallmark of illusory coordination. Three trends emerge: (1) StrategyQA(**CS**)- and MuSiQue(**MH**)-trained topologies maintain positive $\mathcal{O}$ across domains, indicating robust message utilization; (2) CaseHOLD(**L**)-trained topologies show strongly negative $\mathcal{O}$ in most transfers despite high accuracy; and (3) multi-domain training mitigates fluctuations, yielding more stable connections. Refer to Appendix C.6 for further discussion.

| Benchmark | Pearson Correlation | | Ablation Accuracy | | |
|---|---|---|---|---|---|
| | Acc–$\mathcal{R}$ | Acc–$\mathcal{O}$ | In-Domain | Connection-OOD | Role-OOD |
| CaseHOLD | -0.007 | 0.0002 | 63.50 | 62.88 (-0.62) | 48.26 (-15.24) |
| COM$^2$ | -0.035$^{**}$ | 0.045$^{***}$ | 47.90 | 50.68 (+2.78) | 34.50 (-13.40) |
| MuSiQue | 0.003 | 0.123$^{***}$ | 58.40 | 53.04 (-5.36) | 48.44 (-9.96) |
| SciBench | 0.084$^{***}$ | -0.039$^{*}$ | 38.90 | 38.69 (-0.21) | 30.29 (-8.61) |
| TheoremQA | 0.113$^{***}$ | -0.081$^{***}$ | 63.80 | 61.26 (-2.54) | 51.64 (-12.16) |
| StrategyQA | -0.096$^{***}$ | 0.067$^{**}$ | 72.50 | 71.00 (-1.50) | 53.89 (-18.61) |

Table 6: Correlation and ablation results for AgentDropout across 6 datasets. **Acc–$\mathcal{R}$** and **Acc–$\mathcal{O}$** columns report Pearson correlations ($^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$). The right part shows ablation results; parenthesized values indicate accuracy changes (pp). See Appendix C.4 for details.

## 5 Correlation and Ablation Studies

Table 6 reports our correlation and ablation results. In the correlation analysis, we confirm that accuracy is only weakly correlated with $\mathcal{R}$ and $\mathcal{O}$, again highlighting the risk of surface-level evaluation.

In the ablation study, we isolate the contribution of each topological component by fixing one and replacing the other with its OOD counterpart: *Connection-OOD* retains the in-domain roles while replacing the connections with OOD ones; *Role-OOD* does the opposite. Both ablations cause performance drops across all benchmarks except COM$^2$ (**D**), indicating that roles and connections are each susceptible to topological overfitting (§3). Moreover, *Role-OOD* causes a substantially larger average drop than *Connection-OOD* (-13.00 vs. -1.24 percentage points), implying that roles are on average more task-specific than connections.

MuSiQue (**MH**), however, is a notable exception: under *Connection-OOD*, performance drops by 5.36 points, compared with an average of only 0.418 points across the other five datasets. This suggests that, for multi-hop reasoning tasks, valid connections are particularly important.

## 6 Conclusion

This work examines generalization failures in adaptive MAS. We show that adaptive MAS exhibit mixed generalization, succeeding in some settings but failing in others (**topological overfitting**). Going beyond surface-level accuracy, we provide analysis using two new metrics and reveal **illusory coordination**, where strong accuracy masks flawed internal collaboration. These results call for designing adaptive MAS with generalizability in mind and evaluating them through rigorous internal analysis rather than accuracy in isolation.

## Limitations

**Focused Evaluation** To present a generalizable analysis of adaptive MAS, we focused on two representative frameworks: AFlow, which exemplifies constructive topology search, and AgentDropout, which implements dynamic pruning. These methods reflect contrasting design philosophies and our analysis sheds light on topological overfitting and illusory coordination under domain transfer. However, they do not cover the full range of adaptive MAS structures. Other frameworks—potentially using alternative optimization strategies or architectural assumptions—may exhibit different performance profiles. In particular, our scope is centered on adaptive topology-learning MAS in data-scarce settings and does not directly include tool-using or web-surfing agents. Exploring a wider range of adaptive MAS frameworks remains an important direction for future work.

**Extension of Multi-Domain Learning** Our results show that multi-domain training yields notable performance gains compared to topologies optimized for a single dataset. However, our analysis covered only simple, initial attempts within this paradigm; discovering robust methods thus remains an important and promising direction. Such an extension may be essential for shifting the system from task-specific coordination toward more generalized and robust collaborative intelligence.

## Acknowledgments

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2020-II201373, Artificial Intelligence Graduate School Program (Hanyang University)). This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) under the artificial intelligence semiconductor support program to nurture the best talents (IITP-(2026)-RS-2023-00253914) grant funded by the Korea government (MSIT). This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-00558151).

## References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, and 1 others. 2025. Why do multi-agent llm systems fail? *arXiv preprint arXiv:2503.13657*.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022. Lexglue: A benchmark dataset for legal language understanding in english. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 4310–4330.

Ding Chen, Qingchen Yu, Pengyuan Wang, Wentao Zhang, Bo Tang, Feiyu Xiong, Xinchi Li, Minchuan Yang, and Zhiyu Li. 2025. xverify: Efficient answer verifier for reasoning model evaluations. *arXiv preprint arXiv:2504.10481*.

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. 2024. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations*.

Wenhu Chen and 1 others. 2023. Theoremqa: A theorem-driven question answering dataset. *arXiv preprint arXiv:2305.12524*.

Mor Geva and 1 others. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Google. 2025. Gemini 3 flash model card. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Flash-Model-Card.pdf. Accessed: 2025-12-30.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R Bowman, Tim Rocktäschel, and Ethan Perez. 2024. Debating with more persuasive llms leads to more truthful answers. *arXiv preprint arXiv:2402.06782*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Shiyuan Li, Yixin Liu, Qingsong Wen, Chengqi Zhang, and Shirui Pan. 2025. Assemble your crew: Automatic multi-agent communication topology design

via autoregressive graph generation. *arXiv preprint arXiv:2507.18224*.

OpenAI. 2025. Gpt-5.1 instant and gpt-5.1 thinking system card addendum. https://cdn.openai.com/pdf/4173ec8d-1229-47db-96de-06d87147e07e/5_1_system_card.pdf. Accessed: 2025-12-30.

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. 2024. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15174–15186, Bangkok, Thailand. Association for Computational Linguistics.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Chunhao Tian, Yutong Wang, Xuebo Liu, Zhexuan Wang, Liang Ding, Miao Zhang, and Min Zhang. 2025. Agentinit: Initializing llm-based multi-agent systems via diversity and expertise orchestration for effective and efficient collaboration. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 11870–11902.

Harsh Trivedi and 1 others. 2022. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2023. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *arXiv preprint arXiv:2307.10635*.

Zhexuan Wang, Yutong Wang, Xuebo Liu, Liang Ding, Miao Zhang, Jie Liu, and Min Zhang. 2025. Agentdropout: Dynamic agent elimination for token-efficient and high-performance llm-based multi-agent collaboration. *arXiv preprint arXiv:2503.18891*.

Kai Xiong, Xiao Ding, Yixin Cao, Yuxiong Yan, Li Du, Yufei Zhang, Jinglong Gao, Jiaqian Liu, Bing Qin, and Ting Liu. 2025. Com2: A causal-guided benchmark for exploring complex commonsense reasoning in large language models. *arXiv preprint arXiv:2506.07064*.

Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan Suchow, Zhenyu Cui, Rong Liu, and 1 others. 2024. Fincon: A synthesized llm multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. *Advances in Neural Information Processing Systems*, 37:137010–137045.

Guibin Zhang, Luyang Niu, Junfeng Fang, Kun Wang, Lei Bai, and Xiang Wang. 2025. Multi-agent architecture search via agentic supernet. *arXiv preprint arXiv:2502.04180*.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. 2024a. Cut the crap: An economical communication pipeline for llm-based multi-agent systems. *arXiv preprint arXiv:2410.02506*.

Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. 2024b. G-designer: Architecting multi-agent communication topologies via graph neural networks. *arXiv preprint arXiv:2410.11782*.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, and 1 others. 2024c. Aflow: Automating agentic workflow generation. *arXiv preprint arXiv:2410.10762*.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jurgen Schmidhuber. 2024. Gptswarm: Language agents as optimizable graphs. In *Forty-first International Conference on Machine Learning*.

# Appendix

## A Dataset Statistics

Please refer to Table 7. For the training set, we used 60 for AgentDropout and 100 for AFlow.

| Dataset Name (Domain) | Total Train Split | Training Set Size | Test Set Size |
|---|---|---|---|
| CaseHOLD (Legal) | 45,000 | 60 / 100 | 1,000 |
| $COM^2$ (Detective) | 251 | 60 / 100 | 1,004 |
| MuSiQue (Multi-hop) | 1,174 | 60 / 100 | 405 |
| SciBench (Science) | 138 | 60 / 100 | 552 |
| TheoremQA (Math) | 149 | 60 / 100 | 555 |
| StrategyQA (CS) | 2,061 | 60 / 100 | 229 |

Table 7: Dataset statistics across diverse domains. CS denotes Commonsense.

## B Adaptive MAS Algorithms

We summarize the adaptive MAS algorithms used in our experiments and describe their operating mechanisms.

**AFlow** is a topology optimization framework that focuses on automatically discovering the best way for agents to collaborate. Instead of using a predefined communication structure (like a debate or a pipeline), AFlow learns the optimal communication graph from scratch.

The central idea is that the best "workflow" or communication pattern for agents is not known beforehand and should be learned. AFlow takes a constructive approach, starting with nothing and building up the communication graph. The detailed process is described below:

1. Start with Isolated Agents: The process begins with a blank process with a simple 'solve it' prompt.
2. Propose and Evaluate Extensions: The framework iteratively proposes adding new workflows by MCTS.
3. Score and Select: It evaluates how much each potential new workflow improves the system's performance on a given task.
4. Build the Graph: The workflows that provide the most significant performance boost are permanently added to the graph.
5. Iterate: This process repeats, gradually building a complex and effective topology (workflow) optimized for the specific task.

**AgentInit** is an automated MAS *initialization* method that focuses on forming a strong agent team (roles) *before* running the inference framework. Instead of optimizing a communication graph directly, AgentInit first generates a pool of candidate agents, standardizes their role specifications, and then selects a compact, complementary team by jointly balancing *task relevance* and *intra-team diversity*.

The overall procedure is as follows:

1. Multi-round Candidate Generation: A Planner decomposes the user query into sub-tasks and drafts candidate agent roles, while an Observer reviews the decomposition and role assignments and provides feedback. This refinement repeats for multiple rounds.
2. NL-to-Format Standardization: A Formatter converts each candidate agent role from free-form natural language into a standardized representation (e.g., JSON) to ensure consistency for downstream comparison.
3. Construct Candidate Teams: From the candidate agent pool, enumerate possible teams whose size lies within predefined bounds.
4. Score Teams by Relevance and Diversity: Compute a relevance score between each agent (and team) and the query using embedding-based cosine similarity, and measure intra-team diversity using an embedding-similarity matrix (e.g., via Vendi-style diversity).
5. Pareto-based Selection: Identify the Pareto-optimal set of teams that are non-dominated with respect to relevance and diversity, and use a Selector (LLM-powered) to choose the final team for deployment.

**AgentDropout** is another topology optimization framework, but it takes the opposite approach to AFlow. It aims to create a communication structure that is not only high-performing but also token-efficient. Inspired by the "dropout" technique in neural networks, AgentDropout assumes that in any given multi-agent team, some agents or communication links might be redundant or even harmful. It improves performance and efficiency by dynamically pruning or eliminating these non-essential components. The detailed process is described below:

1. Start with a Dense Graph: The process typically begins with a highly-connected graph where most agents can communicate with each other.

2. Identify Redundancy: During different rounds of communication, the framework uses an optimization method to score the importance of each agent and each communication link.

3. Dynamically "Drop" Agents: Agents or links with low importance scores are temporarily dropped out for that round. This forces the system to solve the problem without relying on every single voice, making it more robust.

4. Optimize for Efficiency and Performance: By removing unnecessary communication, the method significantly reduces the number of tokens required, lowering computational costs while often improving the final answer by reducing noise.

# C Additional Experimental Results

## C.1 Cross-Model Comparison on Qwen3-30B-A3B

For completeness, Table 8 reports the corresponding cross-domain generalization results on `Qwen3-30B-A3B`, using the same layout as Table 1.

## C.2 Analysis of Qwen3-30B-A3B Results

**Generality of Topological Overfitting** The results presented in Table 8 demonstrate that the limitations of single-domain optimization observed in the GPT-oss-20B environment also persist in the Qwen3-30B-A3B setting. Specifically, when the AgentDropout algorithm is optimized for the Legal domain, it achieves a solid in-domain performance of 69.7%. However, when this same configuration is transferred to out-of-distribution (OOD) domains such as Detective or Science, the accuracy plummets to 30.6% and 39.1%, respectively. In certain configurations, this performance degradation is even more pronounced than what was observed with GPT-oss-20B. These findings empirically validate that 'topological overfitting' is not a phenomenon restricted to a specific base LLM but is a pervasive challenge across adaptive multi-agent systems (MAS), regardless of the underlying model or the training domain.

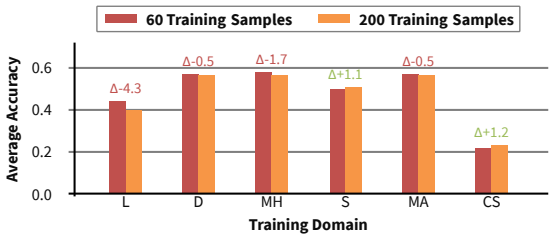



Figure 3: AgentDropout generalization performance under different training-set sizes. Here, generalization performance denotes the mean score across test domains for a fixed training domain, and $\Delta$ denotes the change in this mean score when the training set increases from 60 to 200 examples.

**Quantitative Evidence of Illusory Coordination** While some domain transfer settings might appear successful based on final accuracy, a quantitative analysis of internal system metrics reveals a breakdown in coordination, a phenomenon we term 'illusory coordination'. This is clearly supported by the 'Connection Significance ($\mathcal{O}$)' analysis in Table 9. For instance, when a topology optimized on the Multi-Hop (MuSiQue) domain is transferred, the utility of information exchange between agents—measured by connection significance—recorded negative values across all tested domains, such as -1.00 in the Detective domain and -0.92 in the Commonsense domain. These negative scores indicate that the messages exchanged between agents do not meaningfully contribute to the final answer. This highlights that seemingly high performance in OOD tasks is not a result of generalized MAS coordination, but rather a reliance on the strong standalone reasoning capabilities of the underlying Qwen3 model, masking the failure of the intended multi-agent collaboration.

## C.3 Training Set Size Impact Analysis

Figure 3 provides a complementary view of training-set size for AgentDropout with `GPT-oss-20B`. Increasing the single-domain training budget from 60 to 200 examples yields only modest gains in average held-out accuracy for SciBench (+1.1) and StrategyQA (+1.2). However, the pattern is not consistent: performance declines for $\text{COM}^2$ (-4.3), MuSiQue (-1.7), and both CaseHOLD and TheoremQA (-0.5 each). These mixed results suggest that simply adding more in-domain examples does not reliably improve cross-domain generalization.

| Training / Test (Domain) | L | D | MH | S | MA | CS |
|---|---|---|---|---|---|---|
| CaseHOLD (**L**egal) | **69.8** | 33.9 | 36.8 | 47.8 | 66.1 | 63.5 |
| COM$^2$ (**D**etective) | 69.3 | **40.2** | 24.3 | 13.5 | 29.4 | 14.4 |
| MuSiQue (**M**ulti-**H**op) | 67.8 | 31.8 | **28.0** | 37.2 | 60.5 | 13.2 |
| SciBench (**S**cience) | 69.2 | 29.6 | 19.7 | **34.1** | 61.6 | 34.2 |
| TheoremQA (**M**ath) | 69.5 | 33.7 | 26.5 | 39.5 | **60.7** | 53.0 |
| StrategyQA (**C**ommonsense) | 9.0 | 9.3 | 1.9 | 0.1 | 15.6 | **63.0** |
| Multi-Domain Training | 70.0 | 27.4 | 9.1 | 5.6 | 7.0 | 12.7 |

(a) AgentDropout

| Training / Test (Domain) | L | D | MH | S | MA | CS |
|---|---|---|---|---|---|---|
| CaseHOLD (**L**egal) | **69.7** | 37.0 | 45.4 | 36.0 | 54.8 | 68.9 |
| COM$^2$ (**D**etective) | 66.7 | **38.8** | 49.1 | 42.7 | 63.6 | 79.3 |
| MuSiQue (**M**ulti-**H**op) | 68.6 | 36.8 | **43.6** | 39.3 | 61.3 | 69.4 |
| SciBench (**S**cience) | 69.7 | 30.6 | 38.2 | **39.1** | 59.2 | 61.0 |
| TheoremQA (**M**ath) | 67.8 | 34.5 | 33.7 | 29.7 | **50.9** | 70.0 |
| StrategyQA (**C**ommonsense) | 69.2 | 37.6 | 39.4 | 30.3 | 49.3 | **69.3** |
| Multi-Domain Training | 61.4 | 36.7 | 38.5 | 30.7 | 53.7 | 55.5 |

(b) AFlow

Table 8: In-domain and OOD performance of AgentDropout and AFlow on `Qwen3-30B-A3B`, formatted identically to Table 1. Cell colors are normalized per column by the column-wise maximum: values $\geq 95\%$ max are shaded in blue, and values $< 70\%$ max are in red. This table is provided for cross-model comparison with the main-paper `GPT-oss-20B` results.

| Training / Test (Domain) | L | D | MH | S | MA | CS |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | **1.00** | 0.33 | 0.12 | 0.19 | 0.14 | 0.43 |
| COM$^2$(**D**etective) | 0.22 | **1.00** | 0.23 | 0.24 | 0.28 | 0.37 |
| MuSiQue(**M**ulti-**H**op) | 0.79 | 0.96 | **0.87** | 1.00 | 0.96 | 0.82 |
| SciBench(**S**cience) | 0.82 | 0.83 | 0.80 | **0.99** | 1.00 | 0.93 |
| TheoremQA(**M**ath) | 0.74 | 0.74 | 0.55 | 0.75 | **1.00** | 0.91 |
| StrategyQA(**C**ommonsense) | 0.80 | 0.84 | 0.54 | 0.80 | 0.86 | **1.00** |
| Multi-Domain Training | 0.94 | 0.94 | 0.56 | 0.61 | 0.67 | 1.00 |

| Training / Test (Domain) | L | D | MH | S | MA | CS |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | **1.00** | −0.48 | −0.93 | −0.71 | −0.81 | −0.65 |
| COM$^2$(**D**etective) | −0.01 | **1.00** | −1.21 | −0.61 | −0.60 | −0.35 |
| MuSiQue(**M**ulti-**H**op) | −0.82 | −1.00 | **−0.71** | −0.90 | −0.85 | −0.92 |
| SciBench(**S**cience) | −0.20 | 0.32 | 0.00 | **1.00** | 0.94 | 0.38 |
| TheoremQA(**M**ath) | −0.14 | 0.51 | 0.27 | 1.00 | **0.99** | 0.54 |
| StrategyQA(**C**ommonsense) | 0.36 | 0.75 | 0.96 | −0.01 | −0.13 | **1.00** |
| Multi-Domain Training | 1.00 | 0.99 | −0.81 | 0.90 | 0.93 | 0.45 |

Table 9: Illusory coordination of AgentDropout on `Qwen3-30B-A3B` detected by $\mathcal{R}$ and $\mathcal{O}$. All entries are row-wise normalized by the maximum value in each row (i.e., each cell reports value / max(row)); for the MuSiQue row in the connection metric $\mathcal{O}$, where all entries are negative, row-wise absolute maximum normalization is used instead. Cell colors follow the normalized ratios: values $\geq 0.70$ are shaded blue (i.e., successful transfer), values $< 0.70$ are shaded red (i.e., failed transfer). In-domain results are in bold. Results provide new insights into MAS dynamics.

### C.4 Interchange Ablation

The right block of Table 6 reports an interchange ablation for **AgentDropout** with **GPT-oss-20B**, designed to isolate the relative contribution of roles and connections in the learned topology. In *Connection-OOD*, agent roles are fixed to the test domain while the inter-agent connections are swapped; in *Role-OOD*, the connections are fixed while the agent roles are swapped.

Across most datasets, changing roles under *Role-OOD* causes substantially larger degradation than changing connections under *Connection-OOD*, suggesting that learned role assignments are generally a more performance-critical component of the topology. MuSiQue is a notable exception: under *Connection-OOD*, performance drops by 5.36 points, compared with an average drop of 0.418 points across the other five datasets. This pattern is consistent with the stronger **Acc–$\mathcal{O}$** trend observed for MuSiQue in Table 6, suggesting that connection structure matters more for communication-intensive multi-hop reasoning.

### C.5 Three-Run Results for Cross-Domain Generalization

Adaptive MAS methods can converge to different collaboration topologies due to training data and stochastic optimization. To assess robustness, we therefore report results from three independent train–test runs for each setting. Detailed per-run performance for `GPT-oss-20B` is provided in Table 11, while the corresponding results for `Qwen3-30B-A3B` are provided in Table 12.

### C.6 Three-Run Results for Diagnosing Illusory Coordination

For the same reason, we also report three-run outcomes of our structural diagnostics to verify that *Illusory Coordination* is reproducible rather than seed-specific. Table 13 presents the per-run results for AgentDropout with `GPT-oss-20B` across all transfer settings, while Table 14 reports the corresponding results for AgentDropout with `Qwen3-30B-A3B`.

## D Multi-Agent System Failure Taxonomy

Table 10 summarizes the MAST categories and failure-mode definitions used throughout our anal-

ysis.

To systematically describe error patterns observed in MAS execution traces, we adopt the **Multi-Agent System Failure Taxonomy (MAST)** introduced by Cemri et al., an empirically derived taxonomy built from large-scale analyses of MAS failure traces. MAST organizes failures into **14 failure modes** under three broad categories: *System Design Issues* (violations of task/role specifications or missing termination conditions), *Inter-Agent Misalignment* (breakdowns in coordination such as conversation resets, task derailment, or ignoring other agents' inputs), and *Task Verification* (insufficient or incorrect verification of intermediate/final results). In addition, each failure mode is associated with when it typically emerges in the end-to-end MAS pipeline (pre-execution, execution, and post-execution), highlighting that some issues can span multiple stages and propagate through later interactions.

## E Case Study

Please refer to Figure 4 for the examples.

We present representative qualitative examples showing how adaptive MAS topologies behave under domain transfer. Across cases, we observe (i) **structural collapse**—agents ignore incoming messages or become redundant "copycats," and (ii) **format mismatch**—topologies optimized for binary (T/F) tasks fail on numerical or multi-choice settings. Notably, even when the final answer is correct, traces often reveal weak coordination, suggesting that apparent OOD success may reflect *individual excellence* rather than *effective collaboration*.

## F Prompts

For AFlow, please refer to Figure 5. For AgentDropout, refer to Figure 6. For the prompt for LLM-as-a-Judge used for calculating conection significance score, refer to Figure 7.

The remarkably simple (or 'naive') prompts generated by AFlow when trained on legal and detective domains (as seen in Figure 5) suggest another finding: these seemingly 'underfitted' networks achieve high generalizability without complex inter-agent collaborations, as they rated high generalizability scores. This can also be a reason why we should perform intra-topology diagnosis to reveal true collaboration ability of the topology.

## G Experiment Details

We evaluated both `GPT-oss-20B` (Agarwal et al., 2025) and `Qwen3-30B-A3B` (Team, 2025). Section 3 and Section 4 reports the `GPT-oss-20B` results in the main paper, and Section C.2 reports the corresponding `Qwen3-30B-A3B` results in the appendix. We used vLLM(Kwon et al., 2023) for efficient inference. We ran the model on a single H200 GPU with 140GB of VRAM. Since we utilized the highly parallelizable nature of the vLLM request and ran multiple experiments at once, we couldn't analyze the time taken for a single experiment. The results were gathered with the single run. The parameters used for the experiments are as follows:

- **MASLab**: We configured the MASLab repository for our experiments. First, we added the Qwen3 and GPT-oss model endpoints to compare cross-domain trends across model families. Second, we slightly modified the codebase, which is the change in the topology file path, to make it accept the learned topologies from other datasets.
- **Qwen3-30B-A3B**: We used `Qwen3-30B-A3B` on the vLLM server for the appendix comparison runs. The parameters are shown below:

  –tensor-parallel-size 1

  –gpu-memory-utilization 0.90

  –max-model-len 16384

  –dtype auto

  –max-num-batched-tokens 65536

  –max-num-seqs 512

  –disable-uvicorn-access-log

  –async-scheduling

  –trust-remote-code
- **GPT-oss-20B**: We used `GPT-oss-20B` on the vLLM server for the appendix comparison runs. The parameters are shown below:

  –tensor-parallel-size 1

  –gpu-memory-utilization 0.90

  –max-model-len 16384

  –dtype auto

  –max-num-batched-tokens 65536

  –max-num-seqs 512

| Broad Category | Failure Mode | Description |
|---|---|---|
| System Design Issues | **FM-1.1: Disobey task specification** | Failure to adhere to specified constraints or requirements. |
| | **FM-1.2: Disobey role specification** | Failure to adhere to defined responsibilities of its role. |
| | **FM-1.3: Step repetition** | Unnecessary reiteration of previously completed steps. |
| | **FM-1.4: Loss of conversation history** | Unexpected context truncation, reverting to previous state. |
| | **FM-1.5: Unaware of terminal condition** | Lack of recognition for criteria that triggers interaction end. |
| Inter-Agent Misalignment | **FM-2.1: Conversation reset** | Unexpected restarting of a dialogue, losing progress. |
| | **FM-2.2: Fail to ask clarification** | Inability to request info when faced with unclear data. |
| | **FM-2.3: Task derailment** | Deviation from the intended objective of a given task. |
| | **FM-2.4: Information withholding** | Failure to share important data impacting decision-making. |
| | **FM-2.5: Ignored agent input** | Disregarding input provided by other agents in the system. |
| | **FM-2.6: Reasoning-action mismatch** | Discrepancy between reasoning and actual actions taken. |
| Task Verification | **FM-3.1: Premature termination** | Ending interaction before objectives are met. |
| | **FM-3.2: No/incomplete verification** | Omission of checking of task outcomes or system outputs. |
| | **FM-3.3: Incorrect verification** | Failure to adequately validate information during iterations. |

Table 10: Taxonomy of Multi-Agent System Failure Modes (MAST Framework)

–disable-uvicorn-access-log

–async-scheduling

–trust-remote-code

- **xVerify**: We used LLM-as-a-judge, specifically xVerify-9B-C(Chen et al., 2025) model on vLLM server to determine if the answer is correct or not. The parameters for this model are:

  –tensor-parallel-size 1

  –gpu-memory-utilization 0.8

  –max-model-len 16384

  –trust-remote-code

  –disable-uvicorn-access-log

## H Use of AI Assistants

We used AI assistants to correct grammatical errors and unclear statements in our original writings. We also used coding agent models to write specific scripts for evaluation or data processing.

| Training / Test (Domain) | C | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | 63.9 | 42.7 | 58.5 | 44.2 | 65.6 | 69.0 |
| COM$^2$(**D**etective) | 51.3 | 48.0 | 52.8 | 34.3 | 54.4 | 17.0 |
| MuSiQue(**M**ulti-**H**op) | 64.8 | 50.6 | 58.0 | 40.3 | 65.2 | 74.2 |
| SciBench(**S**cience) | 61.2 | 34.5 | 54.3 | 37.8 | 63.1 | 48.5 |
| TheoremQA(**Ma**th) | 62.2 | 47.1 | 55.8 | 36.1 | 64.0 | 73.8 |
| StrategyQA(**C**ommonsense) | 0.5 | 0.4 | 41.7 | 0.2 | 15.3 | 70.7 |
| Multi-Domain Training | 60.7 | 45.9 | 54.1 | 41.1 | 64.3 | 75.5 |

(a) Run 1: AgentDropout Results

| Training / Test (Domain) | C | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | 61.2 | 39.7 | 45.2 | 41.7 | 65.1 | 69.9 |
| COM$^2$(**D**etective) | 63.5 | 42.9 | 51.9 | 39.9 | 64.9 | 77.3 |
| MuSiQue(**M**ulti-**H**op) | 44.6 | 41.9 | 48.2 | 38.8 | 64.4 | 75.1 |
| SciBench(**S**cience) | 43.3 | 41.8 | 46.9 | 39.1 | 53.0 | 76.9 |
| TheoremQA(**Ma**th) | 39.2 | 39.7 | 44.4 | 29.7 | 55.5 | 69.9 |
| StrategyQA(**C**ommonsense) | 50.3 | 34.0 | 38.8 | 35.7 | 61.6 | 76.4 |
| Multi-Domain Training | 61.4 | 42.2 | 58.8 | 43.8 | 66.9 | 72.5 |

(b) Run 1: AFlow Results

| Training / Test (Domain) | C | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | 64.3 | 48.3 | 56.0 | 41.1 | 66.3 | 70.3 |
| COM$^2$(**D**etective) | 55.8 | 48.3 | 54.6 | 36.1 | 52.8 | 17.0 |
| MuSiQue(**M**ulti-**H**op) | 61.9 | 47.3 | 58.3 | 38.9 | 63.6 | 74.7 |
| SciBench(**S**cience) | 62.4 | 33.5 | 56.0 | 39.2 | 62.7 | 47.6 |
| TheoremQA(**Ma**th) | 62.5 | 47.5 | 59.8 | 36.5 | 64.3 | 74.7 |
| StrategyQA(**C**ommonsense) | 0.7 | 0.5 | 42.0 | 0.2 | 16.2 | 76.0 |
| Multi-Domain Training | 59.2 | 46.5 | 51.4 | 40.9 | 64.5 | 75.1 |

(c) Run 2: AgentDropout Results

| Training / Test (Domain) | C | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | 63.0 | 42.3 | 49.1 | 39.7 | 66.1 | 68.1 |
| COM$^2$(**D**etective) | 62.9 | 42.3 | 47.4 | 37.9 | 56.8 | 21.8 |
| MuSiQue(**M**ulti-**H**op) | 62.4 | 41.4 | 49.1 | 40.0 | 64.7 | 65.9 |
| SciBench(**S**cience) | 62.0 | 37.0 | 48.9 | 32.8 | 61.8 | 71.2 |
| TheoremQA(**Ma**th) | 44.9 | 32.8 | 46.2 | 38.4 | 57.3 | 36.7 |
| StrategyQA(**C**ommonsense) | 34.4 | 41.7 | 41.2 | 40.2 | 67.0 | 76.9 |
| Multi-Domain Training | 62.8 | 42.4 | 46.9 | 39.5 | 65.8 | 74.7 |

(d) Run 2: AFlow Results

| Training / Test (Domain) | C | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | 62.4 | 47.3 | 57.8 | 40.3 | 64.5 | 70.7 |
| COM$^2$(**D**etective) | 62.4 | 47.3 | 57.8 | 40.3 | 64.5 | 70.7 |
| MuSiQue(**M**ulti-**H**op) | 62.9 | 49.2 | 58.8 | 41.2 | 67.4 | 72.5 |
| SciBench(**S**cience) | 61.7 | 34.6 | 54.3 | 39.6 | 62.5 | 46.3 |
| TheoremQA(**Ma**th) | 61.9 | 47.0 | 57.0 | 38.0 | 63.2 | 76.9 |
| StrategyQA(**C**ommonsense) | 0.6 | 0.7 | 40.7 | 0.0 | 15.7 | 70.7 |
| Multi-Domain Training | 60.7 | 47.6 | 53.3 | 41.2 | 64.5 | 75.1 |

(e) Run 3: AgentDropout Results

| Training / Test (Domain) | C | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | 60.9 | 42.2 | 49.4 | 38.8 | 67.9 | 68.6 |
| COM$^2$(**D**etective) | 63.5 | 42.2 | 49.4 | 38.8 | 65.4 | 66.4 |
| MuSiQue(**M**ulti-**H**op) | 48.1 | 40.9 | 46.9 | 39.1 | 56.9 | 5.7 |
| SciBench(**S**cience) | 62.8 | 38.8 | 46.9 | 27.2 | 56.8 | 66.4 |
| TheoremQA(**Ma**th) | 61.5 | 40.7 | 49.6 | 39.7 | 66.7 | 67.7 |
| StrategyQA(**C**ommonsense) | 29.9 | 27.4 | 21.5 | 26.6 | 39.3 | 68.1 |
| Multi-Domain Training | 62.4 | 37.3 | 50.6 | 33.7 | 65.2 | 73.4 |

(f) Run 3: AFlow Results

Table 11: Full raw domain-transfer performance results on `GPT-oss-20B` across three independent runs. Left sub-tables show **AgentDropout** results and right sub-tables show **AFlow** results. Rows indicate the training domain and columns indicate the test domain, ordered as Legal (L), Detective (D), Multi-Hop (MH), Science (S), Math (MA), and Commonsense (C). Multi-Domain Training corresponds to multitask training.

| Training / Test (Domain) | L | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | 70.0 | 33.3 | 35.6 | 46.0 | 68.3 | 63.8 |
| COM$^2$(**D**etective) | 69.4 | 42.1 | 26.0 | 16.8 | 28.3 | 7.4 |
| MuSiQue(**M**ulti-**H**op) | 67.9 | 31.3 | 28.1 | 36.7 | 60.5 | 11.3 |
| SciBench(**S**cience) | 70.0 | 28.9 | 20.7 | 42.3 | 62.3 | 37.6 |
| TheoremQA(**Ma**th) | 68.4 | 33.9 | 26.4 | 40.7 | 59.8 | 46.7 |
| StrategyQA(**C**ommonsense) | 10.8 | 10.0 | 1.7 | 0.2 | 15.5 | 64.5 |
| Multi-Domain Training | 70.1 | 26.5 | 8.1 | 5.5 | 13.5 | 9.2 |

(a) Run 1: AgentDropout Results

| Training / Test (Domain) | L | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | 70.7 | 35.4 | 40.7 | 43.1 | 62.0 | 72.1 |
| COM$^2$(**D**etective) | 69.8 | 39.8 | 47.2 | 43.8 | 63.4 | 87.8 |
| MuSiQue(**M**ulti-**H**op) | 70.3 | 35.5 | 39.3 | 42.6 | 62.3 | 72.5 |
| SciBench(**S**cience) | 70.0 | 35.1 | 42.2 | 43.1 | 63.1 | 72.9 |
| TheoremQA(**Ma**th) | 68.2 | 31.1 | 34.3 | 34.6 | 63.4 | 72.9 |
| StrategyQA(**C**ommonsense) | 68.3 | 40.9 | 54.1 | 36.6 | 51.9 | 70.3 |
| Multi-Domain Training | 46.8 | 35.1 | 41.7 | 36.6 | 64.9 | 74.7 |

(b) Run 1: AFlow Results

| Training / Test (Domain) | L | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | 69.7 | 33.6 | 38.0 | 48.5 | 65.2 | 65.5 |
| COM$^2$(**D**etective) | 69.0 | 38.0 | 24.2 | 17.3 | 29.2 | 5.7 |
| MuSiQue(**M**ulti-**H**op) | 67.1 | 31.3 | 28.4 | 36.3 | 60.9 | 13.1 |
| SciBench(**S**cience) | 68.7 | 28.9 | 18.8 | 42.3 | 62.0 | 32.7 |
| TheoremQA(**Ma**th) | 70.4 | 33.9 | 26.7 | 38.7 | 61.8 | 58.5 |
| StrategyQA(**C**ommonsense) | 8.7 | 8.8 | 2.5 | 0.0 | 16.2 | 61.6 |
| Multi-Domain Training | 70.0 | 27.3 | 8.4 | 5.7 | 11.9 | 5.2 |

(c) Run 2: AgentDropout Results

| Training / Test (Domain) | L | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | 68.3 | 40.5 | 52.8 | 21.9 | 38.9 | 64.6 |
| COM$^2$(**D**etective) | 68.0 | 37.8 | 52.8 | 41.5 | 63.2 | 75.1 |
| MuSiQue(**M**ulti-**H**op) | 69.4 | 37.8 | 53.6 | 40.6 | 61.8 | 73.8 |
| SciBench(**S**cience) | 70.2 | 34.0 | 41.0 | 42.8 | 62.9 | 70.7 |
| TheoremQA(**Ma**th) | 65.2 | 38.7 | 24.0 | 11.1 | 27.0 | 61.1 |
| StrategyQA(**C**ommonsense) | 68.5 | 37.2 | 23.0 | 12.0 | 33.0 | 67.3 |
| Multi-Domain Training | 68.0 | 40.4 | 30.9 | 13.9 | 33.9 | 20.5 |

(d) Run 2: AFlow Results

| Training / Test (Domain) | L | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | 69.7 | 34.8 | 36.8 | 48.7 | 64.9 | 61.1 |
| COM$^2$(**D**etective) | 69.5 | 40.4 | 22.7 | 17.9 | 30.8 | 6.6 |
| MuSiQue(**M**ulti-**H**op) | 68.5 | 32.8 | 27.4 | 38.5 | 60.2 | 15.3 |
| SciBench(**S**cience) | 68.9 | 31.1 | 19.5 | 42.7 | 60.5 | 32.3 |
| TheoremQA(**Ma**th) | 69.8 | 33.3 | 26.4 | 39.2 | 60.4 | 53.7 |
| StrategyQA(**C**ommonsense) | 7.5 | 9.3 | 1.5 | 0.2 | 15.1 | 62.9 |
| Multi-Domain Training | 69.9 | 28.3 | 10.6 | 5.7 | 12.6 | 6.6 |

(e) Run 3: AgentDropout Results

| Training / Test (Domain) | L | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | 70.1 | 35.0 | 42.5 | 42.9 | 63.4 | 69.9 |
| COM$^2$(**D**etective) | 62.4 | 38.7 | 47.2 | 42.8 | 64.1 | 75.1 |
| MuSiQue(**M**ulti-**H**op) | 66.2 | 37.0 | 38.0 | 34.6 | 59.8 | 62.0 |
| SciBench(**S**cience) | 68.8 | 22.7 | 31.6 | 31.3 | 51.5 | 39.2 |
| TheoremQA(**Ma**th) | 69.9 | 33.7 | 42.7 | 43.5 | 62.2 | 76.0 |
| StrategyQA(**C**ommonsense) | 70.7 | 34.7 | 41.2 | 42.4 | 63.1 | 70.3 |
| Multi-Domain Training | 69.5 | 34.7 | 43.0 | 41.5 | 62.3 | 71.2 |

(f) Run 3: AFlow Results

Table 12: Full raw domain-transfer performance results on `Qwen3-30B-A3B` across three independent runs. Left sub-tables show **AgentDropout** results and right sub-tables show **AFlow** results. Rows indicate the training domain and columns indicate the test domain, ordered as Legal (L), Detective (D), Multi-Hop (MH), Science (S), Math (MA), and Commonsense (C). Multi-Domain Training corresponds to multitask training.

| Training / Test (Domain) | L | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | **1.00** | 0.57 | 0.04 | 0.22 | 0.26 | 0.57 |
| COM$^2$(**D**etective) | 0.80 | **1.00** | 0.05 | 0.43 | 0.47 | 0.83 |
| MuSiQue(**M**ulti-**H**op) | 0.66 | 1.00 | **0.39** | 0.45 | 0.54 | 0.60 |
| SciBench(**S**cience) | 0.49 | 0.52 | 0.05 | **1.00** | 0.64 | 0.43 |
| TheoremQA(**M**ath) | 0.38 | 0.37 | 0.05 | 0.64 | **1.00** | 0.36 |
| StrategyQA(**C**ommonsense) | 1.00 | 0.98 | 0.07 | 0.34 | 0.47 | **0.87** |
| Multi-Domain Training | 0.94 | 1.00 | 0.28 | 0.63 | 0.68 | 0.97 |

(a) Run 1: Role Alignment ($\mathcal{R}$) Heatmap

| Training / Test (Domain) | L | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | **1.00** | −0.07 | −1.94 | −2.32 | −2.13 | −1.74 |
| COM$^2$(**D**etective) | 0.84 | **1.00** | 0.13 | 0.46 | 0.41 | 0.21 |
| MuSiQue(**M**ulti-**H**op) | 1.00 | 0.95 | **0.16** | 0.96 | 0.74 | 0.96 |
| SciBench(**S**cience) | −0.79 | −0.53 | −0.57 | **1.00** | 0.86 | −0.01 |
| TheoremQA(**M**ath) | 0.05 | 0.41 | 0.01 | 1.00 | **0.92** | 0.52 |
| StrategyQA(**C**ommonsense) | 0.91 | 0.94 | 0.14 | 1.00 | 0.89 | **0.82** |
| Multi-Domain Training | 0.81 | 0.98 | −0.05 | 1.00 | 0.96 | 0.96 |

(b) Run 1: Connection Significance ($\mathcal{O}$) Heatmap

| Training / Test (Domain) | L | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | **1.00** | 0.56 | 0.04 | 0.22 | 0.26 | 0.53 |
| COM$^2$(**D**etective) | 0.78 | **1.00** | 0.04 | 0.45 | 0.49 | 0.83 |
| MuSiQue(**M**ulti-**H**op) | 0.70 | 1.00 | **0.39** | 0.52 | 0.62 | 0.61 |
| SciBench(**S**cience) | 0.45 | 0.52 | 0.04 | **1.00** | 0.63 | 0.52 |
| TheoremQA(**M**ath) | 0.39 | 0.37 | 0.04 | 0.64 | **1.00** | 0.33 |
| StrategyQA(**C**ommonsense) | 0.90 | 0.83 | 0.06 | 0.27 | 0.37 | **1.00** |
| Multi-Domain Training | 0.87 | 1.00 | 0.33 | 0.58 | 0.61 | 0.86 |

(c) Run 2: Role Alignment ($\mathcal{R}$) Heatmap

| Training / Test (Domain) | L | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | **1.00** | 0.28 | −1.67 | −1.95 | −1.76 | −1.50 |
| COM$^2$(**D**etective) | 0.98 | **0.99** | 0.27 | 1.00 | 0.92 | 0.81 |
| MuSiQue(**M**ulti-**H**op) | 1.00 | 0.99 | **0.15** | 0.92 | 0.81 | 0.71 |
| SciBench(**S**cience) | −0.80 | −0.45 | −0.48 | **1.00** | 0.73 | −0.11 |
| TheoremQA(**M**ath) | −0.49 | −0.07 | −0.19 | 1.00 | **0.88** | 0.28 |
| StrategyQA(**C**ommonsense) | 0.95 | 0.99 | 0.20 | 1.00 | 0.92 | **0.80** |
| Multi-Domain Training | 0.78 | 0.97 | −0.03 | 1.00 | 0.99 | 0.92 |

(d) Run 2: Connection Significance ($\mathcal{O}$) Heatmap

| Training / Test (Domain) | L | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | **1.00** | 0.56 | 0.04 | 0.21 | 0.25 | 0.51 |
| COM$^2$(**D**etective) | 0.78 | **1.00** | 0.05 | 0.40 | 0.45 | 0.80 |
| MuSiQue(**M**ulti-**H**op) | 0.69 | 1.00 | **0.36** | 0.53 | 0.58 | 0.54 |
| SciBench(**S**cience) | 0.40 | 0.43 | 0.04 | **1.00** | 0.59 | 0.43 |
| TheoremQA(**M**ath) | 0.37 | 0.34 | 0.03 | 0.52 | **1.00** | 0.27 |
| StrategyQA(**C**ommonsense) | 1.00 | 0.99 | 0.07 | 0.32 | 0.47 | **0.88** |
| Multi-Domain Training | 0.79 | 0.91 | 0.30 | 0.48 | 0.52 | 1.00 |

(e) Run 3: Role Alignment ($\mathcal{R}$) Heatmap

| Training / Test (Domain) | L | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | **1.00** | −0.04 | −1.80 | −1.99 | −1.85 | −1.48 |
| COM$^2$(**D**etective) | 0.87 | **1.00** | 0.11 | 0.42 | 0.36 | 0.29 |
| MuSiQue(**M**ulti-**H**op) | 1.00 | 0.95 | **0.15** | 0.97 | 0.85 | 0.92 |
| SciBench(**S**cience) | −0.67 | −0.52 | −0.65 | **1.00** | 0.72 | −0.08 |
| TheoremQA(**M**ath) | 0.09 | 0.28 | −0.03 | 1.00 | **0.85** | 0.62 |
| StrategyQA(**C**ommonsense) | 0.92 | 0.92 | 0.19 | 1.00 | 0.89 | **0.82** |
| Multi-Domain Training | 0.26 | 0.86 | −0.78 | 0.33 | 0.28 | 1.00 |

(f) Run 3: Connection Significance ($\mathcal{O}$) Heatmap

Table 13: Per-run (3 independent runs) **Role Alignment** (left) and **Connection Significance** (right) heatmaps for **AgentDropout** with `GPT-oss-20B` under domain transfer across six domains (L, D, MH, S, MA, C; see Table 1). Rows indicate the *training* domain and columns the *test* domain. All entries are *row-wise normalized* by the maximum value in each row (each cell reports $v/\max(\text{row})$, so the row maximum is 1.00). Cells are shaded blue for normalized values $\geq 0.70$ (successful transfer) and red for values $< 0.70$ (failed transfer); intermediate values are left unshaded. Boldface indicates **in-domain** (diagonal) results.

| Training / Test (Domain) | L | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | **1.00** | 0.43 | 0.16 | 0.35 | 0.24 | 0.66 |
| COM$^2$(**D**etective) | 0.23 | **1.00** | 0.23 | 0.32 | 0.36 | 0.33 |
| MuSiQue(**M**ulti-**H**op) | 0.82 | 0.96 | **0.91** | 1.00 | 0.96 | 0.83 |
| SciBench(**S**cience) | 0.81 | 0.83 | 0.80 | **0.90** | 1.00 | 0.94 |
| TheoremQA(**M**ath) | 0.74 | 0.75 | 0.54 | 0.76 | **1.00** | 0.89 |
| StrategyQA(**C**ommonsense) | 0.97 | 0.97 | 0.62 | 0.95 | 1.00 | **0.90** |
| Multi-Domain Training | 0.93 | 0.93 | 0.51 | 0.60 | 0.67 | 1.00 |

(a) Run 1: Role Alignment ($\mathcal{R}$) Heatmap

| Training / Test (Domain) | L | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | **1.00** | 0.09 | −0.92 | −0.78 | −0.85 | −0.67 |
| COM$^2$(**D**etective) | −0.13 | **1.00** | −1.05 | −0.26 | −0.24 | −0.01 |
| MuSiQue(**M**ulti-**H**op) | −0.83 | −1.00 | **−0.75** | −0.90 | −0.86 | −0.93 |
| SciBench(**S**cience) | −0.03 | 0.49 | −0.01 | **1.00** | 0.92 | 0.45 |
| TheoremQA(**M**ath) | −0.15 | 0.48 | 0.28 | 1.00 | **0.99** | 0.46 |
| StrategyQA(**C**ommonsense) | 0.55 | 0.76 | 0.91 | 0.09 | 0.00 | **1.00** |
| Multi-Domain Training | 1.00 | 0.99 | −0.94 | 0.79 | 0.88 | 0.43 |

(b) Run 1: Connection Significance ($\mathcal{O}$) Heatmap

| Training / Test (Domain) | L | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | **1.00** | 0.30 | 0.10 | 0.15 | 0.12 | 0.33 |
| COM$^2$(**D**etective) | 0.19 | **1.00** | 0.20 | 0.12 | 0.16 | 0.40 |
| MuSiQue(**M**ulti-**H**op) | 0.75 | 0.94 | **0.86** | 1.00 | 0.96 | 0.79 |
| SciBench(**S**cience) | 0.82 | 0.82 | 0.80 | **1.00** | 1.00 | 0.92 |
| TheoremQA(**M**ath) | 0.72 | 0.71 | 0.55 | 0.72 | **1.00** | 0.86 |
| StrategyQA(**C**ommonsense) | 0.70 | 0.75 | 0.49 | 0.70 | 0.77 | **1.00** |
| Multi-Domain Training | 0.96 | 0.95 | 0.60 | 0.62 | 0.68 | 1.00 |

(c) Run 2: Role Alignment ($\mathcal{R}$) Heatmap

| Training / Test (Domain) | L | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | **1.00** | −0.29 | −1.03 | −0.89 | −0.94 | −0.81 |
| COM$^2$(**D**etective) | −1.05 | **1.00** | −1.45 | −1.18 | −1.20 | −0.88 |
| MuSiQue(**M**ulti-**H**op) | −0.85 | −1.00 | **−0.75** | −0.91 | −0.87 | −0.95 |
| SciBench(**S**cience) | −0.18 | 0.34 | 0.15 | **1.00** | 0.95 | 0.36 |
| TheoremQA(**M**ath) | 0.04 | 0.65 | 0.27 | 1.00 | **1.00** | 0.72 |
| StrategyQA(**C**ommonsense) | 0.34 | 0.79 | 0.98 | 0.10 | −0.05 | **1.00** |
| Multi-Domain Training | 1.00 | 1.00 | −0.53 | 0.98 | 0.98 | 0.31 |

(d) Run 2: Connection Significance ($\mathcal{O}$) Heatmap

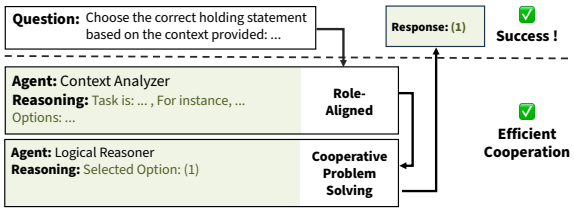
| Training / Test (Domain) | L | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | **1.00** | 0.30 | 0.11 | 0.13 | 0.11 | 0.38 |
| COM$^2$(**D**etective) | 0.25 | **1.00** | 0.25 | 0.30 | 0.35 | 0.39 |
| MuSiQue(**M**ulti-**H**op) | 0.80 | 0.98 | **0.85** | 1.00 | 0.97 | 0.85 |
| SciBench(**S**cience) | 0.77 | 0.80 | 0.75 | **1.00** | 0.95 | 0.89 |
| TheoremQA(**M**ath) | 0.76 | 0.78 | 0.57 | 0.76 | **1.00** | 0.97 |
| StrategyQA(**C**ommonsense) | 0.71 | 0.78 | 0.50 | 0.73 | 0.78 | **1.00** |
| Multi-Domain Training | 0.92 | 0.92 | 0.56 | 0.61 | 0.67 | 1.00 |

(e) Run 3: Role Metric Heatmap

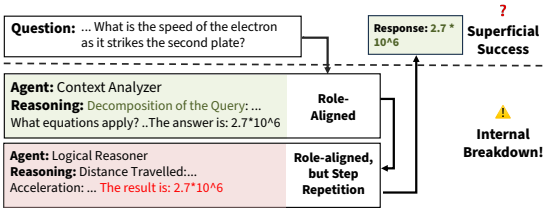
| Training / Test (Domain) | L | D | MH | S | MA | C |
|---|---|---|---|---|---|---|
| CaseHOLD(**L**egal) | **1.00** | 0.16 | −0.85 | −0.43 | −0.63 | −0.46 |
| COM$^2$(**D**etective) | −0.32 | **1.00** | −1.15 | −0.44 | −0.42 | −0.21 |
| MuSiQue(**M**ulti-**H**op) | −0.76 | −1.00 | **−0.64** | −0.90 | −0.82 | −0.89 |
| SciBench(**S**cience) | −0.41 | 0.11 | −0.15 | **1.00** | 0.96 | 0.32 |
| TheoremQA(**M**ath) | −0.26 | 0.42 | 0.26 | 1.00 | **0.99** | 0.48 |
| StrategyQA(**C**ommonsense) | 0.26 | 0.70 | 0.97 | −0.18 | −0.30 | **1.00** |
| Multi-Domain Training | 1.00 | 1.00 | −0.92 | 0.95 | 0.97 | 0.60 |

(f) Run 3: Connection Significance ($\mathcal{O}$) Heatmap

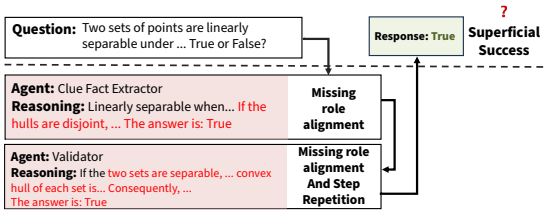
Table 14: Per-run (3 independent runs) **Role Alignment** (left) and **Connection Significance** (right) heatmaps for **AgentDropout** with `Qwen3-30B-A3B` under domain transfer across six domains (L, D, MH, S, MA, C; see Table 1). Rows indicate the *training* domain and columns the *test* domain. All entries are *row-wise normalized* by the maximum value in each row (each cell reports $v/\max(\text{row})$, so the row maximum is 1.00). For the MuSiQue row in **Connection Significance**, where all entries are negative, we instead apply row-wise absolute maximum normalization. Cells are shaded blue for normalized values $\geq 0.70$ (successful transfer) and red for values $< 0.70$ (failed transfer); intermediate values are left unshaded. Boldface indicates **in-domain** (diagonal) results.



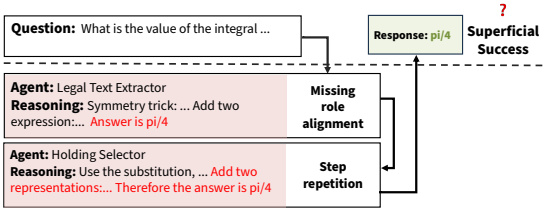

(a) Trained on legal, tested on legal domain



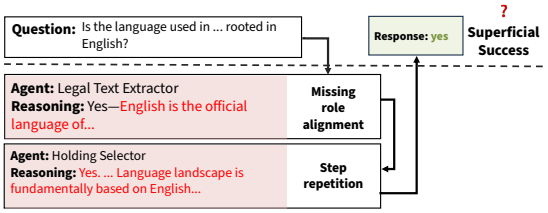

(b) Trained on common sense, tested on science domain




(c) Trained on detective, tested on math domain




(d) Trained on legal, tested on math domain




(e) Trained on legal, tested on common sense domain

Figure 4: Analysis of Illusory Coordination across Six Domain Transfers. While all six cases achieve correct final responses, a stark contrast exists in their internal communication efficiency. Top-Left: The topology optimized for the same domain (ID) exhibits optimal communication efficiency, where agents execute specialized roles with high information gain. Others: In contrast, cross-domain (CD) applications suffer from structural decay; despite correct output, the traces reveal significant role misalignment and communication failure, proving that the results rely on the underlying LLM's priors rather than genuine multi-agent collaboration.

**AFlow Prompt Set (Legal Domain)**

**XXX_PROMPT**
Solve it.

(a) Trained on legal domain

**AFlow Prompt Set (Detective Domain)**

**ANSWER_PROMPT**
Answer the question. Provide the final answer in the format $$((ANSWER))$$. Do not include any other text.

(b) Trained on detective domain

**AFlow Prompt Set (Multihop Domain)**

**PROMPT_SIMPLE**
Answer with ONLY the final answer.

**PROMPT_EXPLAIN**
Provide a brief explanation of how you arrived at the answer, then give the final answer on a new line prefixed by "Answer:".

**PROMPT_CHECK**
Check your answer for correctness. If the answer is a numeric value, replace it with the name of the highest mountain mentioned in the context. If unsure, output "unknown". Provide the answer on a new line prefixed by "Answer:".

**PROMPT_PREFIX**
If the input does not already start with "Answer:", prepend "Answer: " to the input. Return the result.

(c) Trained on multihop domain

**AFlow Prompt Set (Science Domain)**

**BASIC_PROMPT**
Answer the question. Provide the final answer only, enclosed in <answer>. No units, no explanation.

**SOLVE_PROMPT**
Solve the problem step by step. At the end, provide the final answer only, enclosed in <answer>. No units, no explanation.

(d) Trained on science domain

Figure 5: Prompts of topologies optimized by the AFlow algorithm. AFlow dynamically explores the topology via MCTS, so the number of agents and the degree of prompt optimization can vary across datasets (Continued on next page).

**AFlow Prompt Set (Math Domain)**

**PROMPT_ROUND**

You are a math problem solver. Provide the answer as a single answer only. If the answer is an integer, output the integer. If the answer is a decimal, round to one decimal place and output the decimal. Do not include units, explanations, symbols (e.g., √), or scientific notation. Respond with nothing else but the answer.

**PROMPT_ROUND3**

You are a math problem solver. Provide the answer as a single answer only. If the answer is an integer, output the integer. If the answer is a decimal, round to three decimal places and output the decimal. Do not include units, explanations, symbols (e.g., √), or scientific notation. Respond with nothing else but the answer.

**PROMPT_ROUND2**

You are a math problem solver. First, reason step-by-step to verify the result, then provide the final answer as a single answer only.
If the answer is an integer, output the integer. If the answer is a decimal, round to two decimal places and output the decimal. Do not include units, explanations, symbols (e.g., √), or scientific notation. Respond with nothing else but the answer. Remember to double-check the calculation before giving the final answer.

(e) Trained on math domain

**AFlow Prompt Set (Common Sense Domain)**

**MATH_PROMPT**

Answer the following question with only one word

(f) Trained on common sense domain

**AFlow Prompt Set (Multi-task Setting)**

**OPTIMIZED_PROMPT**

Answer the following question. Provide only the final answer in the last line. No additional text. If the answer is numeric, give the number without units. If the answer is a boolean, answer 'True' or 'False'. Do not include any explanation or formatting.

(g) Trained on multitask dataset

Figure 5: Prompts of topologies optimized by the AFlow algorithm. AFlow dynamically explores the topology via MCTS, so the number of agents and the degree of prompt optimization can vary across datasets (continued).

**AgentDropout Prompt Set (Legal domain)**

## LegalQueryInterpreter

You are a **Legal Query Interpreter**. Your task is to analyze a legal text segment containing a hidden citation (marked as <HOLDING> or similar). Identify the **Legal Issue** (e.g., sentencing, jurisdiction) and the **Logical Gap** that the cited case is expected to fill based on the surrounding text.

## HoldingSelector

You are a **Holding Selector**. Your task is to select the most appropriate legal holding from a list of options that matches a specific legal logic or 'gist'. Read the required logic and choose the option that fits best.

## AnswerSynthesizer

"You are an **Answer Synthesizer**. Your task is to take a selection result (an index or an option string) and format it strictly as '(index)'

## LegalTextExtractor

You are **LegalTextExtractor**. Your task is to parse raw legal text to isolate the **Context** (the sentences leading up to the citation) and the **Options** provided.

## Validator

You are **Validator**. Your task is to verify if a selected holding option makes sense within the provided legal context. Check for consistency with the case name (if known) and the flow of the argument.

(a) Trained on legal domain

**AgentDropout Prompt Set (Detective domain)**

## QuestionParser

You are **QuestionParser**, an expert in parsing unstructured text into structured formats. Your task is to extract the 'Question', 'Crime', 'Clue', 'Facts', and 'Options' from the provided text. Ignore the input format source; simply look for these sections and output a valid JSON object.

## ClueFactExtractor

You are **ClueFactExtractor**, an expert in investigative reasoning. Your task is to analyze the provided text (which may be a JSON or raw text) containing a crime, facts, and a question. Extract the **Root Cause** (what actually went wrong) and the **Key Constraints** (facts that limit the possible answers). Output a concise bulleted list of evidence.

## InferenceEngine

You are **InferenceEngine**, an expert in logical deduction. Your task is to evaluate a list of options against a set of evidence/facts. For **EACH** option, determine if it is a 'Valid' or 'Invalid' solution to the problem described. Note that **multiple options can be valid**. Provide a brief reason for each

## AnswerSynthesizer

You are **AnswerSynthesizer**. Your task is to format the final answer based on the provided reasoning or list of valid options. You must output a single line listing the option letters in alphabetical order, separated by commas. Do not add explanations.

## Validator

You are **Validator**, a logic auditor. Your task is to review a proposed answer against the original problem context. Check if the selected options logically follow the facts and if any valid options were missed. Output 'VALID' if the logic holds, otherwise explain the error.

(b) Trained on detective domain

**AgentDropout Prompt Set (Multi-hop domain)**

## CTX-Miner

You are **CTX-Miner**, an expert in Information Retrieval for multi-hop questions. Your goal is to select *only* the paragraphs from the provided context that are necessary to answer the complex query. You must identify the chain of entities (Entity A -> Entity B -> Answer).

## EV-Validator

You are **EV-Validator**, an expert in verifying evidence credibility. Your task is to check if a specific paragraph actually supports a specific sub-question or fact. You must output 'SUPPORTED' or 'NOT_SUPPORTED'.

## EvidenceTracer

You are **EvidenceTracer**, an expert in constructing logical reasoning chains. Your task is to decompose the complex query into steps and link each step to the validated evidence ID. You must resolve variables (e.g., #1, #2) from previous steps.

## AS-Synth

You are **AS-Synth** (Answer Synthesizer). Your task is to formulate a final, concise answer string based on the provided reasoning chain. Do not include the reasoning process in the final output, just the answer entity.

## FC-Check

You are **FC-Check** (Fact Checker). Your task is to verify if the final answer is explicitly present in the source text and answers the original prompt constraints. Return 'VERIFIED' or 'FAILED'.

(c) Trained on multihop domain

**AgentDropout Prompt Set (Scientific reasoning domain)**

## Problem-Interpreter

You are **Problem-Interpreter**. Your task is to parse complex scientific questions to extract the domain, known variables (with units), and the specific target variable to be solved. Format the output as a structured JSON.",

## Equation-Selector

You are **Equation-Selector**. Your task is to identify the scientific principles and select the standard LaTeX equations required to solve the problem based on the extracted variables. List the assumptions if any.",

## Calculator-Agent

You are **Calculator-Agent**. Your task is to perform the mathematical derivation and calculation using the provided equations and variables. Show step-by-step substitution and provide the final numerical result with a verification note.

## Validator-Agent

You are **Validator-Agent**. Your task is to verify the calculated result against physical constraints and dimensional consistency. Ensure the magnitude and sign make sense in the scientific context

## Supervisor

You are **Supervisor**. Your task is to audit the workflow of the agents to ensure the process is logical, complete, and follows instructions. **Do not reveal the final numeric answer** in your output; instead, confirm the integrity of the steps and readiness for final reporting.

(d) Trained on science domain

**AgentDropout Prompt Set (Mathematical reasoning domain)**

## TSP

You are **TSP (Theorem Statement Parser)**. Your goal is to convert a natural language math/science problem into a structured formal representation. Extract the **Given Conditions** (variables, values), the **Goal** (target variable), and the **Domain** (e.g., Calculus, Combinatorics).

## MKRA

You are **MKRA (Math Knowledge Retrieval Agent)**. Your task is to identify and retrieve the specific mathematical theorems, formulas, or definitions required to solve the parsed problem. Explain *why* the theorem applies.

## PSP

You are **PSP (Proof Strategy Planner)**. Your job is to outline a logical, step-by-step strategy to solve the problem using the retrieved theorems. Do not perform the final calculation, but define the execution path.

## FLTP

You are **FLTP (Formalization and Linear Tactic Proof agent)**. Your responsibility is to execute the planned strategy rigorously. Perform the derivations and calculations step-by-step to produce the final result.

## CEV

You are **CEV (Counter-Example Validator)**. Your purpose is to sanity-check the final answer. Verify if the answer type matches (integer vs float), if the magnitude is reasonable, and if any constraints were violated.

(e) Trained on math domain

**AgentDropout Prompt Set (Common sense domain)**

## Context Analyzer

You are a **Context Analyzer**. Your goal is to decode the user's intent by decomposing the StrategyQA query into clear, logical sub-questions. Identify ambiguous entities (e.g., 'The Police' band vs. law enforcement) and the necessary reasoning steps.

## Fact Retrieval Agent

You are a **Fact Retrieval Agent**. Your goal is to gather objective, verifiable facts for specific sub-questions. Do not attempt to answer the main question yet; simply provide the raw data.

## LogicalReasoner

You are the **LogicalReasoner**. Your task is to map the retrieved facts to the original proposition and apply inference rules to derive a truth value. Explain the logical bridge between the facts and the conclusion.

## Answer Synthesizer

You are an **Answer Synthesizer**. Your goal is to format the final response. For StrategyQA, the output must be a strict boolean (True/False) or Yes/No, followed by a concise summary if requested. Keep it minimal.

## QualityAssessor

You are the **QualityAssessor**. Your task is to verify the logical consistency of the derived answer. Ensure that ambiguous terms were correctly disambiguated and that the facts directly support the boolean conclusion.

(f) Trained on common sense domain

**AgentDropout Prompt Set (Multi-task Setting)**

## QuestionCollector

You are QuestionCollector. Your job is to collect a precise question from the user. Respond only with the clarified question; do not provide answers or assumptions.

## ResponseSynthesizer

You are ResponseSynthesizer. Your task is to design a modular plan that covers all required subtasks for the given question.

## FeedbackEvaluator

You are FeedbackEvaluator. Your job is to critique the provided plan, pointing out any gaps or redundancies.

## PlanExecutor

You are PlanExecutor. Execute the plan as outlined, coordinating installed roles and forwarding outputs.

## ResultValidator

You are ResultValidator. Validate the output for correctness, completeness, and format compliance.

(g) Trained on multitask dataset

Figure 6: Prompts of topologies optimized by the AgentDropout algorithm across all training domains.

**LLM-as-a-Judge Prompt**

**LLM-as-a-Judge**
You are evaluating whether information flow between agents in a multi-agent system is **potentially useful**.

## Problem Being Solved
{problem}

## Agent B's Role
{role_description}

## Information Provided by Agent A
{output_from_previous_agent_A}

## Question
Could Agent A's output be **helpful** or **provide context** for Agent B to perform its role?

Consider the following to determine your answer:
- Is the information **broadly relevant** to the problem context?
- Can Agent B **extract any value** from this information, even if it is not a perfect fit?
- Is the information **non-obstructive** (i.e., not completely irrelevant or nonsense)?

Answer with ONLY "Yes" or "No".

Figure 7: LLM-as-a-Judge for identifying connection significance.