Learning Collaborative Reasoning Strategies Through Trust-Weighted Multi-Agent Consensus

Projan Shakya*[1], Kristina Ghimire*[1], Kashish Bataju*[1], Ashwini Mandal[1], Sadikshya Gyawali[1], Manish Awale[1], Manish Dahal[1], Shital Adhikari[1, 2], Sanjay Rijal[1, 3], You Young[4], Vaghawan Ojha[1]

[1]E.K. Solutions Pvt. Ltd., Nepal
[2]Stevens Institute of Technology, USA
[3]Institut de Física d'Altes Energies, Spain
[4]Indiana University East, USA
vpo4@msstate.edu, vaghawan.ojha@ekbana.net

Abstract

Large language models (LLMs) demonstrate strong reasoning capabilities yet often exhibit inconsistency and variability in multi-step or high-stakes tasks, limiting their dependability in autonomous problem solving. To mitigate these challenges, we introduce a multi-agent reinforcement learning (MARL) framework that fosters collaborative reasoning among multiple LLM agents through adaptive strategy selection and reward-guided policy optimization. Within this framework, a Graph Attention Network facilitates strategy selection, while a dynamic trust model prioritizes contributions from reliable agents, promoting both coordination and reasoning diversity. Experimental evaluations on mathematical and scientific reasoning benchmarks reveal substantial performance gains, particularly for smaller models. On GSM-1k, Llama-3.1:8B improves from 65.37% individual to 83.33% consensus accuracy, marking a +17.96% relative gain, while GPT-4.1-nano achieves a +9.54% improvement. Even stronger models such as GPT-4.1-mini exhibit consistent yet moderate boosts across datasets. The findings highlight the effectiveness of trust-aware, reinforcement-driven collaboration in enhancing the accuracy, stability, and robustness of LLM-based reasoning systems.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in reasoning, question answering, and knowledge-intensive tasks. However, individual models often exhibit variability in performance due to inherent biases, limited expertise in specific domains, or uncertainty in difficult problems. To address these limitations, multi-agent systems offer a promising approach by combining the strengths of multiple agents, each contributing complementary reasoning capabilities.

A parallel line of research has explored *prompt engineering* as a resource-efficient approach to enhance reasoning in large language models (LLMs). By embedding task descriptions or exemplars directly into the input, prompting enables models to perform diverse tasks without retraining or fine-tuning. Techniques such as zero-shot and few-shot prompting, and reasoning-oriented strategies like Chain-of-Thought (CoT) (Wei et al. 2023; Tutunov et al. 2024; Besta et al. 2024), improve reasoning by making intermediate thought steps explicit. Extensions including Self-Consistency (CoT-SC) (Wang et al. 2023), which aggregates multiple reasoning paths; Self-Ask (Press et al. 2023), which introduces follow-up questioning; Decomposition (Patel et al. 2022), which structures problems into subtasks; and Least-to-Most (Zhou et al. 2023), which solves simpler components sequentially, further expand reasoning capabilities. However, these methods still rely on static prompt formats and lack adaptability to dynamic or multi-agent reasoning contexts, motivating the need for more flexible and adaptive frameworks.

In this work, we propose MARL-GAT (Multi Agent Reinforcement Learning-Graph Attention Network), a trust-aware multi-agent reasoning framework that integrates reinforcement learning with Graph Attention Networks (GATs) (Veličković et al. 2018) to enable adaptive, reliable coordination among language model agents. The system dynamically models trust relationships by maintaining a trust matrix that evolves based on historical performance and reliability. While the GAT processes graph connectivity to enable collaborative reasoning, trust relationships are incorporated through the consensus mechanism, which assigns higher weights to more trustworthy agents based on their historical performance. This allows the system to prioritize contributions from more reliable agents while still incorporating insights from all agents in the final decision-making process. Reinforcement learning optimizes strategy selection across the network, ensuring that agents choose reasoning strategies based on their policies.

We evaluate the framework across benchmark datasets using GPT-based models (OpenAI 2023), including GPT-4.1-mini and GPT-4.1-nano, and observe consistent improvements under trust-aware consensus aggregation. Smaller models such as Llama-3.1:8B (Grattafiori et al. 2024) show

*These authors contributed equally.

the largest relative gains, demonstrating that our method particularly benefit weaker systems by correcting individual errors and enhancing collective reliability. These results confirm that the proposed architecture delivers robust reasoning outcomes across both larger and smaller models (Table 1).

In this work, we make the following key contributions:

- We apply trust-aware multi-agent reinforcement learning framework that enables collaborative reasoning among multiple LLM agents, enhanced with a Graph Attention Network to model trust and prioritize reliable contributions.
- We introduce adaptive strategy selection and reward mechanisms that balance individual performance, consensus alignment, and diverse reasoning approaches to enhance decision-making and exploration.
- Our method improves collaboration, leading to steady gains in consensus accuracy across different model sizes and benchmarks, demonstrating its effectiveness regardless of the underlying model architecture.

2 Related Works

Graph Neural Networks (GNNs) provide a framework for learning over graph-structured data, enabling agents or nodes to exchange information with neighbors (Scarselli et al. 2009). Among these, Graph Attention Networks (GATs) (Veličković et al. 2018) introduce self-attention mechanisms to assign different weights to neighboring nodes, allowing more reliable or informative agents to exert greater influence. Unlike spectral-based GNNs, GATs avoid costly matrix operations and are naturally applicable to inductive settings. GATs have achieved state-of-the-art results across transductive and inductive benchmarks, including citation networks such as Cora, Citeseer, and Pubmed, as well as protein-protein interaction networks where test graphs are unseen during training. By integrating GATs into MARL frameworks, it becomes possible to model trust and dynamically adjust agent influence, leading to more robust consensus and collaborative reasoning.

Consensus-based aggregation has been widely studied as a means to improve collective decision-making. Traditional approaches, such as majority voting or weighted voting, combine agent outputs to enhance overall accuracy (Zhang, Yang, and Başar 2021). Recent works incorporate agent reliability, performance history, or network structure to weigh contributions dynamically. For instance, reinforcement learning-based trusted consensus mechanisms enable agents to independently decide which neighbors to communicate with, effectively handling unreliable agents and improving consensus success rates (Fung et al. 2024). In the context of large language model (LLM) agents, trust-aware consensus mechanisms further improve reasoning robustness by prioritizing reliable agents while preserving diversity, effectively mitigating errors from individual models with lower confidence or expertise.

Multi-agent systems powered by large language models (LLMs) have recently emerged as a promising paradigm for complex reasoning and simulation tasks. For instance, LLMs can replace traditional agent programs in simulations such as ant colony foraging or bird flocking, enabling more flexible and adaptive coordination (Jimenez-Romero, Yegenoglu, and Blum 2025). (Jia et al. 2025) and (Wan et al. 2025) further explore graph-based MARL settings in which a central node decomposes a task into sub-tasks that neighboring agents solve independently, and (Jia et al. 2025) also highlights the role of intrinsic and extrinsic rewards in shaping agent behavior.

3 Dataset

To evaluate both mathematical and scientific reasoning, we employ three complementary benchmark datasets: **ARC-Challenge**, **GSM8k**, and **GSM1k**.

The ARC-Challenge dataset (Clark et al. 2018) consists of grade-school science questions designed to assess reasoning beyond factual recall. These questions span multiple reasoning types, such as causal, comparative, and hypothetical reasoning, making the dataset well-suited for evaluating scientific and commonsense inference capabilities. For this benchmark, 200 questions from the training split are used for model optimization, and evaluation is performed on 300 randomly sampled test questions to ensure representative performance assessment.

For mathematical reasoning, we use the GSM8k dataset (Cobbe et al. 2021), a collection of grade-school mathematical word problems with detailed step-by-step solutions. GSM8k spans arithmetic, algebra, and multi-step reasoning, providing a rich source of structured examples for supervision.

We evaluate generalization on GSM1k (Zhang et al. 2024), a held-out test set designed to mirror the style of GSM8k while containing entirely new problem instances. To prevent data contamination and reduce overfitting, we avoid using GSM1k during training. Instead, our training set consists of 200 problems randomly sampled from GSM8k, ensuring that the model learns general reasoning patterns rather than memorizing benchmark questions.

Overall, this dataset suite offers balanced coverage across reasoning domains: ARC-Challenge for scientific and commonsense reasoning, GSM8k for structured mathematical problem-solving, and GSM1k for generalization to unseen math problems. We selected these benchmarks because they provide precise, unambiguous ground-truth answers, allowing reward signals to remain noise-free and directly computable. Since all problems come with well-defined solutions, evaluation does not rely on subjective human judgment, enabling our MARL framework to operate with perfectly accurate rewards and supporting a clean, interference-free assessment of reasoning performance

4 Methodology

The architecture of the proposed multi-agent reasoning framework, depicted in Figure 1, delineates the end-to-end reasoning pipeline from input embedding to consensus-based answer synthesis. In the initial phase of the reasoning pipeline, input questions are converted into dense embeddings that encapsulate their semantic representations. These

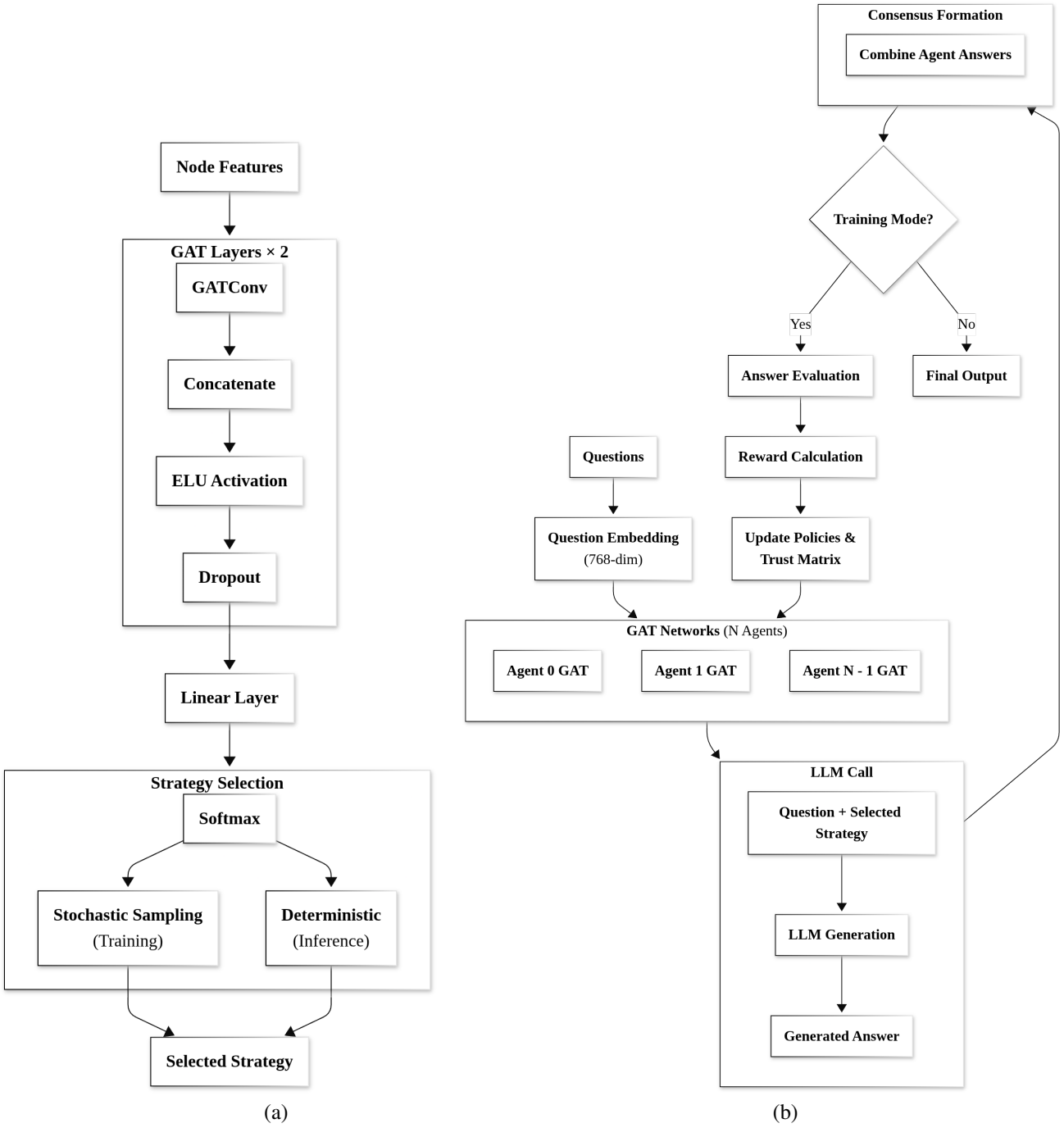

Figure 1: Overview of the MARL-GAT framework. (a) GAT-based policy network for reasoning strategy selection. (b) End-to-end processing workflow illustrating multi-agent reasoning and strategy coordination.

embeddings guide the selection of reasoning strategies, determining how the task should be allocated and approached by the agents. The question is first analyzed to determine an appropriate strategy, after which it is distributed across multiple agents, each utilizing its associated large language model (LLM) to generate a candidate answer. The collection of answers from the agents is then processed by a consensus mechanism, which aggregates the responses to derive a unified output.

The consensus stage also serves a dual purpose by informing the computation of rewards, which measures the quality and reliability of individual agent contributions. These rewards feed into the policy and trust update component, enabling the system to iteratively adjust its internal models of

agent reliability and effectiveness. This adaptive update directly influences subsequent strategy selection, closing the feedback loop that governs system improvement over time. Ultimately, the consensus-driven result is released as the system output, representing the collective reasoning of the agents.

4.1 Setup

The multi-agent system initialization involves four key components: the N-agent architecture, GAT policy networks, strategy effectiveness tracking, and trust matrix setup.

Multi-Agent Architecture The system initializes N autonomous agents, where N represents the total number of agents. Each agent $i \in \{0, 1, \ldots, N-1\}$ has an independent GAT-based policy network $\pi_i(\cdot)$ (where π_i is the policy function for agent i) and an experience memory buffer that stores state–action–reward tuples (s, a, r). Each agent maintains a separate Adam optimizer instance, all initialized using the same learning rate α.

GAT Policy Network Agents communicate over a small-world graph $G = (V, E)$ (Watts and Strogatz 1998) with initial degree k, rewiring probability p, and dynamic evolution constrained by minimum and maximum agent degrees. Each agent uses a Graph Attention Network (GAT) policy with input dimension d_{in}, hidden dimension d_{hidden}, h attention heads, and output dimension d_{out} corresponding to reasoning strategies. The architecture comprises two GAT layers followed by a softmax output:

$$\pi = (\mathbf{W} \cdot \mathrm{GAT}_2(\mathrm{GAT}_1(\mathbf{X}, \mathbf{E}), \mathbf{E})) \tag{1}$$

where $\mathbf{X}$ is the node feature matrix, $\mathbf{E}$ is the edge index tensor, $\mathbf{W}$ is the output weight matrix, and π is the logits obtained from policy network. The GAT layers operate on the node feature matrix $\mathbf{X}$ and edge index tensor $\mathbf{E}$ to produce node embeddings. These embeddings are passed through a fully connected layer parameterized by the weight matrix $\mathbf{W}$, which outputs the raw logits corresponding to the available reasoning strategies.

Strategy Effectiveness Tracking The system tracks effectiveness for the provided reasoning strategies (Expained later in Section 4.8). Scores are updated via exponential moving average with α (EMA smoothing factor).

Trust Matrix The trust matrix $\mathbf{T} \in \mathrm{R}^{N \times N}$ (where $\mathbf{T}$ represents inter-agent trust relationships) is initialized with $T_{i,j} = 0.5$ for connected agents (where $T_{i,j}$ is the trust from agent i to agent j) and $T_{i,i} = 0$ (no self-trust). Trust evolves through symmetric updates between agent pairs, where trust changes are calculated using case-specific base rates with diminishing returns factors that reduce update magnitude as trust levels increase. Updates are applied bidirectionally between connected agents based on their individual correctness relative to the group consensus.

4.2 Question Embedding

We generate 768-dimensional question embeddings using the all-mpnet-base-v2 Sentence Transformer model (Reimers and Gurevych 2019) to capture their semantic content. These embeddings serve as shared inputs for all agents in the multi-agent system and are used both by the policy network for strategy selection and for clustering semantically similar questions to update the strategy effectiveness tracker.

4.3 Cosine Similarity for Adaptive Strategy Selection

While the mechanisms above capture graph-structured reasoning preferences and the diversity of strategies across questions, further refinement is achieved by incorporating semantic similarity between them. Cosine similarity is used to compare the embedding of the current question with those of previously encountered questions, allowing the system to identify semantically related cases. Historical effectiveness scores from these similar questions are then aggregated using a weighted scheme in which closer matches exert greater influence, thereby refining the bias term that is added to the raw logits generated by the GAT. This integration enables the strategy selection process to benefit not only from neural inference and strategy diversity, but also from accumulated experience on related problem types. When no sufficiently similar questions are found, the system defaults to neutral effectiveness estimates, ensuring robustness even in scenarios involving novel or out-of-distribution inputs.

4.4 Strategy Selection

The strategy selection process begins by encoding each question into a dense embedding that is broadcast to all participating agents. Although each agent receives the same question representation, their decision-making diverges due to differences in their positions within the communication graph and the local attention dynamics of the GAT. Each agent's policy network employs multi-head attention to aggregate contextual signals from neighboring nodes, refining its internal representation of the question based on local dependencies and trust-weighted connections.

From the GAT policy network, raw logits over available reasoning strategies are produced and subsequently refined using a *strategy effectiveness tracker*. This tracker introduces a bias toward strategies that have historically yielded higher success rates on similar questions, promoting the reuse of empirically strong approaches. The bias term is added to the raw logits (prior to softmax normalization) and weighted by a tunable coefficient to balance exploration and exploitation. The final strategy for each agent is then sampled from a categorical distribution derived from the adjusted logits, ensuring that while all agents begin from the same question embedding, their ultimate strategy choices remain diverse and informed by both structural and performance-based signals.

Formally, the probability of selecting a strategy is given by

$$P(\text{strategy}) = \text{softmax}\Big(\mathrm{GAT}_{\text{output}} + \gamma \times S_{s,q}^{(t)}\Big) \tag{2}$$

where γ is a tunable parameter controlling the influence of the effectiveness bias $S_{s,q}^{(t)}$, as described in Section 4.8.

4.5 LLM Inference

After an agent selects a reasoning strategy through the combined GAT policy network and strategy tracker, the framework retrieves the corresponding prompt and queries an LLM to produce the final answer. The process involves prompt selection, model inference, and answer extraction. Each strategy uses a distinct prompt template tailored to elicit its specific reasoning behavior. Model inference employs both private and local LLMs: GPT-4.1-mini and GPT-4.1-nano via OpenAI API (OpenAI 2023), and a locally hosted 4-bit quantized LLaMA 3.1-8B model (Grattafiori et al. 2024).

4.6 Consensus

Our multi-agent system employs a weighted consensus mechanism to aggregate agent responses into a collective decision. This mechanism incorporates agent trust to determine each agent's influence on the final outcome, while also guiding reward assignment and trust updates. Agents with higher trust naturally exert greater influence during consensus, yet contributions from lower-trust agents are never discarded, ensuring that diverse reasoning signals remain part of the decision-making process and allowing previously unreliable agents the opportunity to recover

Each agent i is assigned a weight defined as:

$$W_{agent} = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} T_{ij}, \tag{3}$$

where $\mathcal{N}_i = \{j \mid T_{ij} > 0,\, j \neq i\}$ denotes the set of agents trusted by agent i.

For each candidate answer, the total score is computed as the sum of weights of agents selecting that answer:

$$S_{\text{answer}} = \sum_{\text{agent} \in P_{\text{answer}}} W_{\text{agent}} \tag{4}$$

The final consensus is determined by applying a softmax function over these scores to produce a probability distribution:

$$P(\text{answer}_i) = \frac{\exp(S_{\text{answer}_i})}{\sum_j \exp(S_{\text{answer}_j})} \tag{5}$$

where P_{answer} denotes the set of agents choosing that answer. The answer with the highest probability is selected as the consensus answer, and the corresponding probability serves as a confidence measure.

This consensus mechanism ensures that agents with higher trust have greater influence, while still allowing collective input from all agents. The resulting probabilities not only provide the final decision but also enable uncertainty quantification, which can inform downstream processes such as reward assignment, performance tracking, and trust updates between agents.

4.7 Rewards

The reward mechanism forms the core feedback signal driving the learning dynamics of the MARL-GAT framework. In this study, each agent is trained to optimize its reasoning performance through an individual reward scheme, where reinforcement is determined solely by the correctness of its generated output. This direct and interpretable signal encourages agents to focus on producing accurate solutions rather than relying on external or consensus-based influences. The reward function is formally defined as:

$$R_{\text{individual}} = \begin{cases} +1.0, & \text{if correct,} \\ -0.5, & \text{if incorrect.} \end{cases} \tag{6}$$

This formulation provides a clear learning objective for each agent, reinforcing correct reasoning traces while penalizing deviations that lead to incorrect conclusions. By restricting the feedback to an individual level, the learning process remains interpretable and avoids potential instability introduced by inter-agent dependencies. This design allows each policy to evolve independently, ensuring that the improvement in reasoning accuracy arises from the agent's own adaptive behavior rather than collective bias.

During training, the individual reward directly modulates the policy gradient updates through the GAT policy network, influencing strategy selection probabilities and decision-making patterns. As training progresses, the reward-driven updates enable agents to refine their strategy preferences and progressively align their reasoning approaches with ground-truth solutions. This mechanism forms the foundation upon which the overall MARL-GAT learning pipeline operates, supporting consistent and reproducible improvement across multiple reasoning benchmarks.

4.8 Updates

The multi-agent system employs multiple update mechanisms that drive learning and adaptation across different timescales. After a series of interactions, individual rewards are calculated, strategy effectiveness scores are updated based on performance, and trust relationships are adjusted between agents. Policy gradient updates refine agent strategies based on accumulated rewards, while graph structure evolution adapts the communication topology according to trust patterns. This multi-scale approach ensures that both immediate correctness signals and long-term performance patterns are incorporated into the evolution of agent policies, strategy evaluations, and trust relationships. Figure 2 illustrates how these processes interact with the N-agent architecture through the centralized consensus mechanism.

After all agents provide their answers and the consensus mechanism determines the final decision, three complementary update processes are triggered.

GAT Policy Updates Each agent's GAT policy is updated using the REINFORCE algorithm (Williams 1992), prioritizing simplicity. Individual rewards are assigned based on answer correctness, as described in Section 4.7. The stored log probabilities are combined with the reward signals to

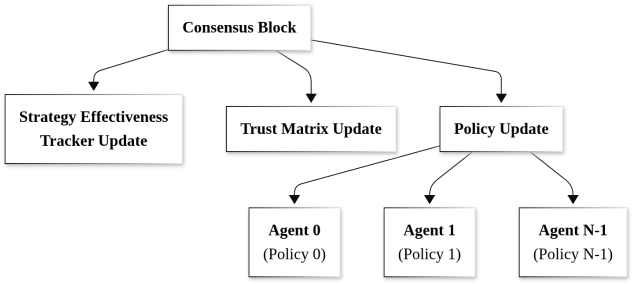

Figure 2: Multi-agent system update architecture showing N agents feeding into a consensus block, which triggers three parallel update processes: GAT policy updates, strategy effectiveness updates, and trust matrix updates.

compute policy gradients, which are then applied to update the network parameters. In this way, each agent gradually adapts its decision-making policy to maximize expected long-term reward.

Strategy Effectiveness Updates In addition to individual policy adaptation, the system maintains a single global tracker that records the effectiveness of each reasoning strategy using question embedding similarity. This tracker is updated using an exponential moving average controlled by a smoothing factor α, allowing both stability and adaptability.

$$S_{s,q}^{(t+1)} = (1-\alpha)S_{s,q}^{(t)} + \alpha \cdot c_{s,q}^{(t)} \tag{7}$$

where $S_{s,q}$ denotes the effectiveness score for strategy s on question embedding q at time t, and $c_{s,q}^{(t)}$ is the binary correctness indicator (1 if correct, 0 if incorrect). When selecting strategies, agents query this shared knowledge base using cosine similarity to find historically effective strategies for similar questions, enabling them to exploit collective learning patterns.

Trust Matrix Update Mechanisms Beyond local learning and global strategy evaluation, the system maintains a dynamic trust network that governs the strength of connections between agents. Trust values evolve continuously based on both immediate correctness signals and longer-term performance outcomes, ensuring that collaboration patterns reflect reliability.

Answer-Based Trust Updates: After each question, the trust between every pair of agents is updated according to their correctness outcomes. If both agents are correct (TT), their mutual trust increases by a base amount δ_T, with diminishing returns applied as trust values approach the upper bound. If one agent is correct and the other is incorrect (TF or FT), trust decreases symmetrically by a smaller amount δ_D, reflecting divergent reliability. If both are incorrect (FF), trust decreases by the largest penalty δ_P, discouraging the reinforcement of shared mistakes. Trust values are bounded between predefined minimum and maximum thresholds to prevent extreme values. All updates are applied symmetrically, such that for agents i and j, trust is updated as $T_{ij} \leftrightarrow T_{ji}$. This symmetric update mechanism ensures that trust relationships evolve based on consistent performance patterns while maintaining balanced influence between agents over time.

Graph Structure Evolution: At fixed intervals, the underlying graph structure is adapted to reflect the evolving trust distribution. This structural evolution proceeds in four stages:

1. Low-trust connections (< 0.3) are removed to prevent reinforcement of unreliable links.
2. High-trust connections (> 0.8) are added to strengthen collaboration between reliable agents.
3. Minimum connectivity is enforced to avoid agent isolation.
4. Existing links are rebalanced by replacing lower-trust connections with higher-trust ones to better reflect the current trust distribution

This structural plasticity allows the network topology itself to emphasize meaningful collaboration pathways.

Trust Matrix Synchronization: Whenever the graph structure changes, the trust matrix is immediately synchronized to ensure numerical consistency with the updated connectivity. This guarantees coherence between the structural and quantitative representations of trust.

Overall Flow and Properties: Together, these mechanisms create a layered update process: answer-based trust updates occur after every question, graph structure evolution takes place periodically, and trust matrix synchronization ensures structural alignment. The resulting trust network is characterized by symmetry ($trust_{i,j} = trust_{j,i}$), boundedness ($0.1 \leq trust \leq 0.95$), diminishing returns at higher trust levels, and strong performance-dependence. Moreover, its structure-aware adaptation guarantees that graph topology evolves in tandem with agent reliability, producing a trust network that is both resilient and aligned with collective effectiveness.

5 Results

5.1 Benchmark Performance

We evaluate our method on two benchmarks: ARC-Challenge, testing scientific and commonsense reasoning, and GSM1k, focusing on step-by-step mathematical problem solving. Table 1 summarizes zero-shot, individual agent, and consensus accuracies, with the "Improvement" column quantifying the benefits of consensus aggregation. Across both benchmarks, consensus consistently enhances performance, reducing individual variability and stabilizing predictions.

On ARC-Challenge, consensus provides moderate gains. GPT-4.1-mini improves from 95.17% to **97.00%** (+1.83%), GPT-4.1-nano rises from 88.80% to **92.00%** (+3.20%), and LLaMA-3.1 8B jumps from 79.20% to **86.00%** (+6.80%). These results indicate that for scientific and commonsense reasoning, consensus acts primarily as a stabilizer.

For GSM1k, consensus has a stronger effect in numerical reasoning. GPT-4.1-mini improves from 90.73% to **94.00%** (+3.27%), GPT-4.1-nano from 83.13% to **92.67%** (+9.54%), and LLaMA-3.1 8B from 65.37% to **83.33%**

Model	Zero-Shot	Individual Accuracy	Consensus Accuracy	Improvement
ARC-Challenge				
GPT-4.1-mini	94.33%	95.17%	**97.00%**	**+1.83%**
GPT-4.1-nano	91.67%	88.80%	**92.00%**	**+3.20%**
Llama-3.1:8B	84.67%	79.20%	**86.00%**	**+6.80%**
GSM-1k				
GPT-4.1-mini	93.33%	90.73%	**94.00%**	**+3.27%**
GPT-4.1-nano	90.00%	83.13%	**92.67%**	**+9.54%**
Llama-3.1:8B	77.33%	65.37%	**83.33%**	**+17.96%**

Table 1: Performance comparison of different models on individual and consensus accuracy across datasets.

Learning Paradigm	Individual Accuracy	Consensus Accuracy
Supervised Learning	83.06%	92.00%
Reinforcement Learning	**83.13%**	**92.67%**

Table 2: Comparison of Supervised Learning and Reinforcement Learning on GSM1K using GPT-4.1-nano.

(+17.96%). This demonstrates that collaborative decision-making is particularly effective in error-prone domains, correcting individual mistakes and substantially enhancing reliability.

Consensus improves accuracy across zero-shot outputs, with moderate gains for strong models and substantial improvements for weaker or error-prone models. Practical considerations, such as parsing outputs only within `<answer>` tags, may slightly affect measured accuracy. Overall, reinforcement learning enables adaptive coordination, and the GAT framework ensures trust-aware consensus, together delivering measurable gains in accuracy, robustness, and stability across diverse reasoning tasks.

5.2 Ablation Study

Reinforcement vs. Supervised Learning To evaluate the impact of the learning paradigm on agent coordination and reasoning performance, we conducted ablation studies on the GSM1K dataset using GPT-4.1-nano as the underlying language model. In the Supervised Learning (SL) setting, agents are trained to predict correct answers by comparing the consensus-derived output with the ground truth from the dataset, optimizing for immediate accuracy. In the Reinforcement Learning (RL) setting, agents learn policies through reward-driven feedback, adapting strategies based on individual correctness and interactions with other agents over multiple episodes. The comparison isolates the effect of the learning paradigm while keeping all other experimental conditions identical.

The results indicate that RL improves both individual accuracy (83.13% vs. 83.06%) and consensus accuracy (92.67% vs. 92.00%), suggesting that reward-driven adaptation enhances coordination and collective reasoning even under similar base conditions. Considering the expense of GPT-based API calls and the more gains from RL, we adopted the RL paradigm for subsequent experiments, balancing performance improvement with cost-efficiency.

Consensus-Aware Reward Integration To evaluate the impact of collaborative feedback on agent performance, we introduce a consensus-aware reward enhancement that extends the baseline individual reward formulation. Unlike the purely correctness-based scheme used in the main model, this approach adjusts rewards based on agreement patterns among agents, aiming to promote both independence and coordinated reasoning.

Formally, the final reward for each agent is defined as:

$$R_{\text{modified}} = R_{\text{individual}} + \Delta R, \tag{8}$$

where,

$$\Delta R = \begin{cases} +\delta^{+}, & \text{if consensus is incorrect and agent is correct,} \\ -\delta^{-}, & \text{if consensus is incorrect and agent is incorrect,} \\ +\delta^{\text{minor}}, & \text{if consensus is correct and agent is in the correct minority,} \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

To examine the impact of consensus-sensitive reward modifications, we compare performance under the standard individual reward mechanism (base reward) with the previously proposed consensus-aware reward scheme. The consensus-aware reward adjusts agent feedback based on alignment with group consensus, rewarding agents who are correct despite an incorrect consensus, penalizing agents who incorrectly conform, and giving minor incentives to correct minority contributors.

Table 3 summarizes consensus accuracy for both reward strategies across the ARC-Challenge and GSM-1k benchmarks. For each model, consensus performance is shown under base rewards versus consensus-aware rewards.

The results indicate that, across both benchmarks, the base individual rewards generally achieve higher consensus accuracy than the consensus-aware modifications. While the consensus-aware reward was designed to encourage alignment and independent correctness, in practice, it introduces slight constraints that can reduce performance in deterministic or low-ambiguity tasks. The ablation highlights that, for these reasoning benchmarks, simpler base rewards are sufficient to guide agents effectively, providing more stable consensus outcomes without the additional complexity of consensus-sensitive adjustments.

Model	Base Reward	Consensus-Aware Reward
ARC-Challenge		
GPT-4.1-mini	**97.00%**	96.33%
GPT-4.1-nano	92.00%	**92.33%**
Llama-3.1:8B	86.00%	86.00%
GSM-1k		
GPT-4.1-mini	**94.00%**	93.67%
GPT-4.1-nano	**92.67%**	92.00%
Llama-3.1:8B	**83.33%**	82.67%

Table 3: Consensus Accuracy Comparison: with Base rewards and Consensus-Aware Reward.

Strategy Weight (γ)	Consensus Accuracy (%)	Individual Accuracy (%)
0.0	89.67	67.16
0.1	**92.67**	83.13
0.7	90.33	**87.46**

Table 4: Impact of Strategy Effectiveness Weight on GSM1K Performance using GPT-4.1-nano.

Strategy Effectiveness Bias Analysis To evaluate the impact of historical strategy performance on agent decision-making, we conducted ablation studies on the GSM1K dataset using GPT-4.1-nano as the underlying language model. The effectiveness bias was applied to the raw logits from the GAT policy network, where $\gamma = 0.0$ represents pure model-based strategy selection without historical guidance, and higher values indicate stronger reliance on empirically successful strategies.

Table 4 summarizes the results. At $\gamma = 0.0$, the system achieved a consensus accuracy of 89.67% and individual agent accuracy of 67.16%, reflecting limited coordination due to the absence of historical feedback. Introducing a moderate effectiveness bias of $\gamma = 0.1$ led to the highest consensus accuracy of **92.67%**, while maintaining a strong individual accuracy of 83.13%. In contrast, a high bias value of $\gamma = 0.7$ produced the best individual accuracy of **87.46%**, but resulted in a lower consensus performance (90.33%), likely due to premature convergence on dominant strategies and reduced exploratory behavior. These findings suggest that moderate historical guidance (specifically $\gamma = 0.1$) provides the best trade-off between strategy diversity and consensus reliability.

6 Future Enhancements

Future work will focus on enhancing the scalability, adaptability, and strategy learning of MARL-GAT. The graph could dynamically adjust its size based on question complexity, and agents could generate or refine strategies informed by prior performance. Cross-domain transfer of trust patterns and strategy effectiveness could reduce retraining needs.

Integrating heterogeneous LLM agents of varying sizes, architectures, or expertise can improve consensus by combining complementary strengths and mitigating correlated errors, with dynamic assignment based on historical reliability. Hierarchical attention mechanisms and dynamic strategy creation could further enhance adaptability to novel tasks.

Advances in trust modeling, considering reasoning quality, explanation clarity, and efficiency, along with temporal tracking and adaptive thresholds, can guide collective decisions more effectively. These directions aim to make MARL-GAT more scalable, interpretable, and robust for complex multi-agent reasoning tasks.

7 Discussion and Conclusion

Our MARL-GAT system demonstrates the effectiveness of combining multi-agent coordination with graph-based trust networks for complex reasoning tasks. Consistent gains in consensus accuracy over individual agents confirm that collaborative reasoning through dynamic trust relationships enhances problem-solving capabilities. The trust network evolves from random initialization to an optimized structure, showing agents learn to maintain connections with high-performing collaborators.

Our framework shows consistent improvements across benchmarks and model sizes. On GSM1K benchmark , GPT-4.1-mini achieves 94%, surpassing GPT-4o (92.9%) and GPT-4 (92.3%), while GPT-4.1-nano reaches 92.67%, and LLaMA-3.1 8B improves from 69.0% to 83.33% (Zhang et al. 2024). On the ARC benchmark, GPT-4.1-mini attains 97%, exceeding GPT-4 (96.4%) and nearly matching LLaMA-3.1 405B (96.9%), with GPT-4.1-nano at 92% and LLaMA-3.1 8B rising to 86% from 83.4% (HyperAI 2025). These results demonstrate that our approach effectively enhances reasoning performance for both large and mid-sized language models.

While stronger GPT-based models mainly benefit from stabilizing consensus, smaller models like LLaMA-3.1 8B achieve the largest relative improvements, indicating that collective agreement is especially valuable for weaker systems. The dynamic graph architecture preserves agent diversity while fostering collaboration, and strategy effectiveness tracking allows adaptive selection of reasoning approaches, contributing to robustness. Limitations include reliance on pre-trained models, which may propagate biases, and increased computational cost from using multiple agents.

Overall, multi-agent reinforcement learning with graph attention networks significantly improves reasoning performance through structured collaboration. The framework achieves measurable gains across datasets and demonstrates broader applicability to other reasoning and knowledge-intensive tasks, though practical deployment requires careful consideration of cost and scalability.

Acknowledgments

We would like to express our sincere gratitude to E.K. Solutions Pvt. Ltd. (EKbana Nepal) for generously providing the time, resources, and support necessary to conduct this research.

References

Besta, M.; Blach, N.; Kubicek, A.; Gerstenberger, R.; Podstawski, M.; Gianinazzi, L.; Gajda, J.; Lehmann, T.; Niewiadomski, H.; Nyczyk, P.; and Hoefler, T. 2024. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16): 17682–17690.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1955–1967. Association for Computational Linguistics.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*.

Fung, H. L.; Darvariu, V.-A.; Hailes, S.; and Musolesi, M. 2024. Trust-based Consensus in Multi-Agent Reinforcement Learning Systems. arXiv:2205.12880.

Grattafiori, A.; Dubey, A.; Jauhri, A.; and et al. 2024. The LLaMA 3 Herd of Models. arXiv:2407.21783.

HyperAI. 2025. Common Sense Reasoning On ARC Challenge. Accessed: 2025-10-27.

Jia, Z.; Li, J.; Qu, X.; and Wang, J. 2025. Enhancing Multi-Agent Systems via Reinforcement Learning with LLM-based Planner and Graph-based Policy. arXiv:2503.10049.

Jimenez-Romero, C.; Yegenoglu, A.; and Blum, C. 2025. Multi-Agent Systems Powered by Large Language Models: Applications in Swarm Intelligence. arXiv:2503.03800.

OpenAI. 2023. GPT API. https://platform.openai.com/. Accessed: 2025-08-18.

Patel, P.; Mishra, S.; Parmar, M.; and Baral, C. 2022. Is a Question Decomposition Unit All We Need? arXiv:2205.12538.

Press, O.; Zhang, M.; Min, S.; Schmidt, L.; Smith, N. A.; and Lewis, M. 2023. Measuring and Narrowing the Compositionality Gap in Language Models. arXiv:2210.03350.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1): 61–80.

Tutunov, R.; Grosnit, A.; Ziomek, J.; Wang, J.; and Bou-Ammar, H. 2024. Why Can Large Language Models Generate Correct Chain-of-Thoughts? arXiv:2310.13571.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. arXiv:1710.10903.

Wan, Z.; Li, Y.; Wen, X.; Song, Y.; Wang, H.; Yang, L.; Schmidt, M.; Wang, J.; Zhang, W.; Hu, S.; and Wen, Y. 2025. ReMA: Learning to Meta-think for LLMs with Multi-Agent Reinforcement Learning. arXiv:2503.09501.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. arXiv:2203.11171.

Watts, D. J.; and Strogatz, S. H. 1998. Collective dynamics of 'small-world' networks. *Nature*, 393(6684): 440–442.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; and Zhou, D. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4): 229–256.

Zhang, H.; Da, J.; Lee, D.; Robinson, V.; Wu, C.; Song, W.; Zhao, T.; Raja, P.; Zhuang, C.; Slack, D.; Lyu, Q.; Hendryx, S.; Kaplan, R.; Lunati, M.; and Yue, S. 2024. A Careful Examination of Large Language Model Performance on Grade School Arithmetic. arXiv:2405.00332.

Zhang, K.; Yang, Z.; and Başar, T. 2021. Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms. arXiv:1911.10635.

Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q.; and Chi, E. 2023. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. arXiv:2205.10625.