FEDGOTA: TOWARDS EFFECTIVE FEDERATED MULTI-AGENT COLLABORATION VIA GRAPH OPTIMAL-TRANSPORT ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

In multi-agent systems (MAS) powered by large language models (LLMs), communication topologies orchestrate interactions among specialized agents, enabling coordinated reasoning on complex tasks. However, learning these topologies is often coupled with task supervision, making the resulting structures task-specific and difficult to transfer to new tasks. In federated settings, this coupling can further confine clients to task-specific knowledge silos, hindering effective aggregation and cross-client knowledge transfer. To address these limitations, we decouple structure learning from task supervision by introducing a shared structural self-supervised objective. Building on this formulation, we propose **FedGOTA**, a federated multi-agent collaboration framework via graph optimal-transport alignment. On each client, several complementary collaboration views are fused to train a topology generator with the self-supervised objective, allowing it to produce reusable, question-aware collaboration structures without task labels. At the server, we probe client topology generators with shared semantic anchors to construct graph codebooks, whose optimal-transport alignment guides personalized aggregation to mitigate negative transfer across heterogeneous clients. Extensive experiments on six reasoning benchmarks across four task domains show that FedGOTA achieves competitive performance compared with fourteen baselines spanning single-agent reasoning and multi-agent collaboration.

1 INTRODUCTION

Multi-agent systems (MAS), powered by large language models (LLMs), coordinate specialized agents through structured communication, supporting applications in complex reasoning (Du et al., 2023), software engineering (Hong et al., 2024), scientific discovery (Swanson et al., 2025), and decision support (Kim et al., 2024).

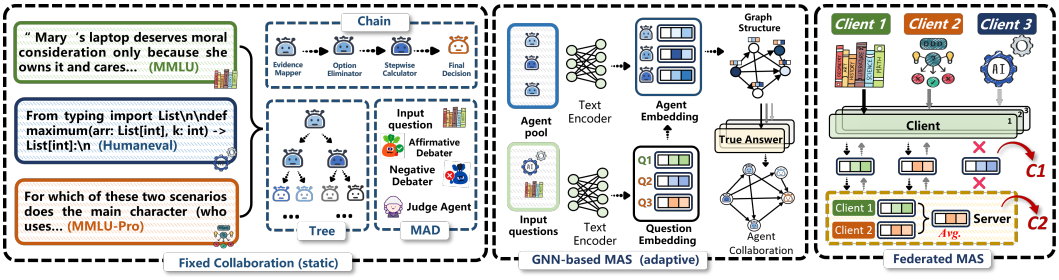


Figure 1: Illustration of multi-agent collaboration paradigms.

MAS collaboration approaches are generally categorized into *static* and *adaptive* regimes, governed by whether the underlying collaboration topology dynamically evolves with the task input. Static methods rely on predefined candidate experts and communication workflows (Li et al., 2023; Hong et al., 2024; Qian et al., 2024), whereas adaptive methods learn to construct question-aware collaboration structures from task-specific data (Zhang et al., 2024; Li et al., 2026a;b). As shown in Fig. 1, static methods offer cross-task reusability, yet their fixed collaboration structures limit adaptability

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

to heterogeneous task requirements. Adaptive approaches transfer the semantics of questions and expert prompts into the graph domain, thereby formulating communication topology design as a graph structure learning problem. Despite their flexibility, adaptive methods still rely on task-specific supervision to optimize collaboration topologies. This dependence can favor collaboration patterns tailored to the training tasks, limiting their reuse across heterogeneous task settings.

These limitations call for a unified collaboration paradigm that combines cross-task knowledge reuse with question-aware coordination. Learning such knowledge from diverse task distributions, however, becomes more challenging when privacy constraints prevent client data from being pooled for centralized training. Fortunately, federated learning (FL) offers a natural alternative, enabling clients to share collaboration knowledge while keeping their data local (Yan et al., 2026; Zhu et al., 2026). However, standard federated aggregation alone does not decouple topology learning from task-specific supervision. Task-specific knowledge silos may therefore persist, hindering the reuse of collaboration knowledge across heterogeneous clients. This naturally gives rise to our first challenge:

C1: *How can we learn reusable collaboration structures across heterogeneous clients without relying on task-specific supervision?* As illustrated in the middle panel of Fig. 1, existing topology generators rely on task-specific rewards for parameter optimization. In federated settings, such task-coupled structures prevent knowledge transfer across clients addressing different tasks owing to discrepancies in model parameters and reward signals. As shown in the right panel of Fig. 1, clients 1 and 2 exchange knowledge through the server, whereas task differences prevent client 3 from participating in aggregation, confining it to local updates.

To move beyond such task-wise isolation, we seek to decouple topology learning from task-specific supervision, enabling collaboration knowledge to be shared across heterogeneous clients. Even with this decoupling, however, client parameters alone do not provide the server with a reliable basis for assessing structural affinity, making it difficult to balance distinct collaboration preferences and transfer knowledge effectively. This leads to our second challenge:

C2: *How can we reliably assess structural affinity across clients to guide personalized transfer of collaboration knowledge?* As shown in the right panel of Fig. 1, clients 1 and 2 may favor distinct collaboration patterns even when addressing the same task. The server, however, lacks a shared semantic reference for reliably assessing their structural affinity. Without such guidance, uniform aggregation may average out useful local collaboration preferences and induce negative transfer. Furthermore, restrictions on raw-data sharing leave the server without direct evidence for reliably estimating inter-client affinity scores.

To address these challenges, we revisit the optimization objective of multi-agent collaboration and develop a unified federated formulation that decouples topology learning from task-specific supervision while enabling personalized knowledge transfer. Building on this formulation, we propose FedGOTA, a framework for personalized federated multi-agent collaboration via graph optimal transport alignment. On each client, we integrate complementary collaboration views to train a question-aware topology generator under a shared structural self-supervised objective. To capture distinct collaboration preferences across clients, shared semantic anchors are introduced at the server to construct graph codebooks. An optimal transport problem is formulated over these codebooks to quantify structural affinity across clients and guide personalized aggregation of topology-generator parameters. After federated training, the personalized generators can be directly reused to construct collaboration structures for downstream reasoning without further optimization.

Our main contributions are outlined as follows:

- We revisit the optimization objective of adaptive multi-agent collaboration and formulate a generalized federated MAS paradigm that enables cross-task reuse of collaboration knowledge through unified graph self-supervision.
- We propose a multi-view structural self-supervised learning scheme that derives supervision from complementary collaboration views, enabling heterogeneous clients to learn reusable collaboration patterns under a unified objective without task-specific answers.
- We propose a server-side aggregation mechanism that quantifies structural affinity via optimal transport over client graph codebooks, enabling selective knowledge transfer across heterogeneous tasks.

2 PRELIMINARIES

Multi-agent collaboration for an input question q is represented as a directed graph $\mathcal{G}_q = (\mathcal{A}, \mathcal{E}_q)$, where $\mathcal{A} = \{a_i\}_{i=1}^N$ denotes the set of candidate agents. The edge set $\mathcal{E}_q$ encodes directed information flow, with $(a_i, a_j) \in \mathcal{E}_q$ indicating communication from agent a_i to agent a_j. A shared text encoder $\mathcal{T}(\cdot)$ maps the question q and the role prompt p_i of agent a_i into a common semantic space:

$$\mathbf{e}_q = \mathcal{T}(q), \qquad \mathbf{e}_i = \mathcal{T}(p_i). \tag{1}$$

The candidate agent embeddings are collected in $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_N]^\top \in \mathbb{R}^{N \times d}$, where d denotes the embedding dimension. A topology generator Γ_Θ with learnable parameters Θ constructs a question-aware collaboration topology from these semantic representations:

$$\mathbf{A}_q = \Gamma_\Theta(\mathbf{e}_q, \mathbf{E}), \tag{2}$$

where $\mathbf{A}_q$ denotes the weighted adjacency matrix of $\mathcal{G}_q$, with $[\mathbf{A}_q]_{ij}$ specifying the communication weight from agent a_i to agent a_j. For notational simplicity, we write $\Gamma_\Theta(q, \mathcal{A})$ for $\Gamma_\Theta(\mathbf{e}_q, \mathbf{E})$, with the shared text encoding implicitly included. Guided by $\mathbf{A}_q$, the agents exchange messages and produce a final prediction:

$$\hat{y} = \mathcal{F}_{\mathrm{MAS}}(q, \mathcal{A}, \mathbf{A}_q), \tag{3}$$

where $\mathcal{F}_{\mathrm{MAS}}$ denotes the multi-agent reasoning process governed by the collaboration topology, and $\hat{y}$ is the resulting task prediction.

3 REVISITING AND GENERALIZING MULTI-AGENT COLLABORATION

In this section, we revisit the optimization objective of adaptive multi-agent collaboration and develop a generalized federated formulation for heterogeneous client tasks.

Given a task dataset $\mathcal{D} = \{(q, y)\}$, where y denotes the task-specific supervision signal, the topology generator is conventionally optimized as:

$$\Theta^* = \arg\min_{\Theta} \mathbb{E}_{(q,y)\sim\mathcal{D}} \left[\mathcal{L}_{\mathrm{task}} \left(\mathcal{F}_{\mathrm{MAS}}\big(q, \mathcal{A}, \Gamma_\Theta(q, \mathcal{A})\big), y \right) \right], \tag{4}$$

where $\mathcal{L}_{\mathrm{task}}$ evaluates the multi-agent output against y and provides task-specific feedback for topology optimization.

Unfortunately, the objective in Eq. 4 tightly couples topology learning with task-specific supervision, limiting the reuse of learned collaboration knowledge across heterogeneous tasks. More critically, this dependence ties local topology learning to client-specific task objectives, thereby hindering the establishment of a common structural learning criterion for federated collaboration. To overcome this limitation, we reformulate the task-dependent collaboration objective into a generalized federated formulation without task-specific supervision.

Definition 3.1 (Generalized Collaboration Objective)**.** For M clients with heterogeneous local distributions $\{\mathcal{D}_m\}_{m=1}^M$, the objective is to learn client-specific topology generators by minimizing a shared graph self-supervised loss across clients:

$$\{\Theta_m^*\}_{m=1}^M = \arg \min_{\{\Theta_m\}_{m=1}^M} \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}_{q\sim\mathcal{D}_m} \left[\mathcal{L}_{\mathrm{ss}} \left(q, \mathcal{A}, \Gamma_{\Theta_m}(q, \mathcal{A}) \right) \right], \tag{5}$$

where $\mathcal{L}_{\mathrm{ss}}$ provides structural supervision without task-specific answers, and Θ_m denotes the topology-generator parameters of client m.

Definition 3.1 establishes a common structural learning criterion across heterogeneous clients while allowing their collaboration topologies to adapt to individual questions. Because collaboration preferences remain influenced by local task distributions, uniform averaging of the Θ_m would erase useful client-specific patterns. We therefore let the server perform personalized aggregation according to structural affinity:

$$\widetilde{\Theta}_m = \sum_{n=1}^{M} \mathcal{K}(p_m, p_n)\, \Theta_n^*, \tag{6}$$

where p_m denotes the collaboration representation of client m, and $\mathcal{K}(p_m, p_n)$ is an aggregation weight determined by the structural affinity between clients m and n, with $\mathcal{K}(p_m, p_n) \geq 0$ and $\sum_{n=1}^{M} \mathcal{K}(p_m, p_n) = 1$.

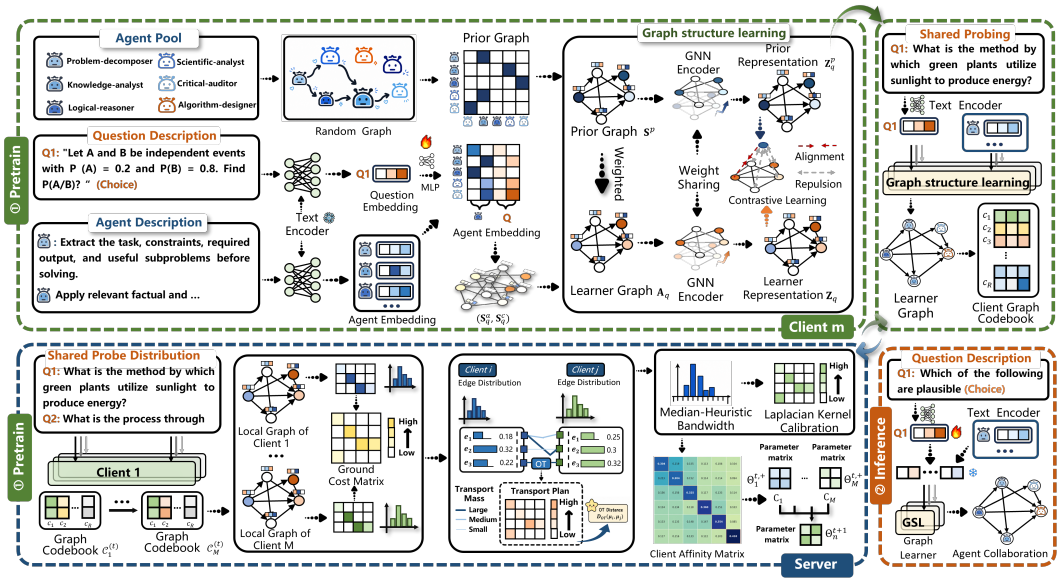


Figure 2: The Framework of FedGOTA

4 METHOD

In this section, we present FedGOTA, which instantiates the generalized federated objective of Section 3 by letting clients learn reusable collaboration structures through multi-view self-supervision and letting the server aggregate generators according to structural affinity (Figure 2).

4.1 MULTI-VIEW COLLABORATION STRUCTURE LEARNING

For the generalized federated objective in Eq. 5, the challenge lies in deriving shared structural supervision while preserving the flexibility of question-aware collaboration. The key insight is that a shared structural reference can guide collaboration learning across tasks while allowing expert interactions to vary with the question. This motivates a multi-view formulation of the topology generator Γ_{Θ_m} through three complementary views: question-aware collaboration, task affinity, and a shared structural prior.

Question-Aware Collaboration View. Different questions require different interaction patterns. To characterize each expert in the context of the current question, we couple its role embedding with question semantics:

$$\mathbf{x}_{q,i} = [\bar{\mathbf{e}}_i \parallel \bar{\mathbf{e}}_i \odot \bar{\mathbf{e}}_q], \tag{7}$$

where $\bar{\mathbf{e}} = \mathbf{e}/\|\mathbf{e}\|_2$ denotes the normalized embedding, with $\parallel$ representing concatenation and $\odot$ element-wise multiplication. The resulting representations, collected in $\mathbf{X}_q = [\mathbf{x}_{q,1}, \ldots, \mathbf{x}_{q,N}]^\top$, are adapted to the local task distribution through a client-specific MLP $\mathcal{F}_{\theta_m}$:

$$\mathbf{H}_{m,q} = \mathcal{F}_{\theta_m}(\mathbf{X}_q, \bar{\mathbf{e}}_q). \tag{8}$$

Since the influence of one expert on another need not be symmetric, separate source and target projections are introduced to characterize directed interactions:

$$\mathbf{S}^c_{m,q} = \frac{(\mathbf{H}_{m,q}\mathbf{W}^s_m)(\mathbf{H}_{m,q}\mathbf{W}^t_m)^\top}{\sqrt{d_h}}, \tag{9}$$

where $\mathbf{W}^s_m$ and $\mathbf{W}^t_m$ are learnable source and target projection matrices, respectively, and d_h denotes the projection dimension.

Task-Affinity View. Complementing the learned inter-expert relations, task affinity explicitly captures the semantic relevance of each expert to the current question. With the normalized expert embeddings collected in $\bar{\mathbf{E}} = [\bar{\mathbf{e}}_1, \ldots, \bar{\mathbf{e}}_N]^\top$, the question–expert affinity vector is defined as:

$$\mathbf{c}_q = \exp\left(\frac{\bar{\mathbf{E}}\bar{\mathbf{e}}_q}{\tau_a}\right), \tag{10}$$

where the exponential is applied element-wise, and $\tau_a > 0$ is a temperature parameter controlling the contrast between affinity scores. To express task relevance at the interaction level, each expert pair is assigned the sum of the affinity scores of its two endpoints:

$$\mathbf{S}_q^a = \mathbf{c}_q \mathbf{1}^\top + \mathbf{1}\mathbf{c}_q^\top, \tag{11}$$

where $\mathbf{1} \in \mathbb{R}^N$ denotes the all-ones vector.

Shared Structural Prior View. To provide a common reference for learning these question-dependent structures, we introduce a task-independent sparse directed acyclic graph $\mathcal{G}^p = (\mathcal{A}, \mathcal{E}^p)$ shared across clients. Motivated by the effectiveness of random collaboration topologies (Qian et al., 2025), we sample a fixed prior once and share it across all clients. It provides a common structural reference that does not depend on any task data, allowing different clients to optimize under the same criterion. Formally, the shared prior is represented by a binary adjacency matrix:

$$\mathbf{S}^p = [\mathbb{I}((a_i, a_j) \in \mathcal{E}^p)]_{i,j=1}^N, \tag{12}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function, yielding $\mathbf{S}^p \in \{0,1\}^{N \times N}$.

Hierarchical Fusion of Collaboration Structures. To jointly capture learned expert interactions and explicit task relevance, the two question-dependent views are fused through channel-wise attention:

$$\mathbf{D}_{m,q} = \mathcal{M}_{\phi_m}\left(\mathbf{S}_{m,q}^c, \mathbf{S}_q^a\right), \tag{13}$$

where $\mathcal{M}_{\phi_m}$ adaptively balances the contributions of the two views, with ϕ_m denoting its learnable parameters.

To balance question-specific adaptation with shared structural guidance, we incorporate the prior through weighted interpolation:

$$\mathbf{A}_{m,q} = \Pi_{\mathcal{G}}((1-\alpha)\mathbf{D}_{m,q} + \alpha \mathbf{S}^p), \tag{14}$$

where $\alpha \in [0,1]$ balances the contributions of the question-aware structure and the shared prior, while $\Pi_{\mathcal{G}}$ applies structural constraints followed by row normalization.

4.2 Unified Structural Self-Supervision

To derive a common learning signal that requires no task labels, we align the representations of corresponding experts under $\mathbf{A}_{m,q}$ and $\mathbf{S}^p$. Specifically, the same graph encoder $\mathcal{G}_{\Omega_m}$ maps the question-aware features $\mathbf{X}_q$ to expert representations under the two structural views:

$$\left(\mathbf{Z}_{m,q}, \mathbf{Z}_{m,q}^p\right) = \mathcal{G}_{\Omega_m}(\mathbf{X}_q, (\mathbf{A}_{m,q}, \mathbf{S}^p)). \tag{15}$$

Although the prior topology is fixed, the question semantics embedded in $\mathbf{X}_q$ shape how the shared structure is represented. Expert identity establishes a natural correspondence across views: representations of the same expert form positive pairs, while those of different experts serve as negatives. These correspondences guide structural learning via a bidirectional contrastive objective:

$$\mathcal{L}_{m,q}^c = -\frac{1}{2N}\sum_{i=1}^N \left[\log \frac{\exp(s_{ii})}{\sum_{j=1}^N \exp(s_{ij})} + \log \frac{\exp(s_{ii})}{\sum_{j=1}^N \exp(s_{ji})}\right], \tag{16}$$

where s_{ij} measures cross-view agreement through a temperature-scaled cosine similarity:

$$s_{ij} = \frac{\left\langle \mathbf{z}_{m,q,i}, \mathbf{z}_{m,q,j}^p \right\rangle}{\tau_c \|\mathbf{z}_{m,q,i}\|_2 \|\mathbf{z}_{m,q,j}^p\|_2}, \tag{17}$$

where $\mathbf{z}_{m,q,i}$ and $\mathbf{z}_{m,q,j}^p$ denote expert representations from the learned and prior views, respectively, and $\tau_c > 0$ is the contrastive temperature. The topology generator and auxiliary graph encoder are jointly optimized by minimizing the expected contrastive loss over local questions:

$$\min_{\Theta_m, \Omega_m} \mathbb{E}_{q \sim \mathcal{D}_m}\left[\mathcal{L}_{m,q}^c(\Theta_m, \Omega_m)\right], \tag{18}$$

where $\Theta_m = \{\theta_m, \mathbf{W}_m^s, \mathbf{W}_m^t, \phi_m\}$ collects the topology-generator parameters, while Ω_m denotes the parameters of the auxiliary graph encoder. The auxiliary encoder remains local, while the topology-generator parameters $\Theta_m^{t,+}$ obtained after local optimization in round t are uploaded to the server for personalized aggregation.

4.3 STRUCTURE-AWARE PERSONALIZED FEDERATION

Although topology learning follows a shared self-supervised objective, collaboration preferences remain shaped by local task distributions, motivating personalized aggregation guided by structural affinity. To assess this affinity under matched question semantics, client-specific topology generators are probed at the server with shared semantic anchors, each comprising multiple paraphrases of the same question. The shared collection of R semantic anchors is denoted by:

$$\mathcal{Q}^{\mathrm{sa}} = \{\mathcal{Q}_r^{\mathrm{sa}}\}_{r=1}^{R}, \tag{19}$$

where $\mathcal{Q}_r^{\mathrm{sa}} = \{q_r^{(v)}\}_{v=1}^{V}$ denotes the group of V paraphrases associated with semantic anchor r. The influence of probe construction is further examined in Appendix A.10.

The structural responses to each anchor group are consolidated into a graph codeword, with the resulting collection forming a client-specific graph codebook:

$$\mathcal{C}_m^{(t)} = \left\{ \frac{1}{V} \sum_{v=1}^{V} \Gamma_{\Theta_m^{t,+}}\left(q_r^{(v)}, \mathcal{A}\right) \right\}_{r=1}^{R}, \tag{20}$$

where each codeword $\mathbf{C}_{m,r}^{(t)}$ is obtained by element-wise averaging the graph responses to the V paraphrases of anchor r.

The key insight is that similar collaboration preferences may be expressed through different yet semantically related expert interactions, motivating an optimal transport formulation for comparing graph codebooks. Accordingly, the transport cost between directed interactions is defined by the semantic discrepancies between their respective source and target experts:

$$\mathbf{G}_{(i,j),(u,v)} = \|\bar{\mathbf{e}}_i - \bar{\mathbf{e}}_u\|_2^2 + \|\bar{\mathbf{e}}_j - \bar{\mathbf{e}}_v\|_2^2. \tag{21}$$

To compare collaboration preferences under the same semantic anchor, we solve the entropy-regularized optimal transport problem:

$$\mathbf{P}_\varepsilon^* = \underset{\mathbf{P} \in \Pi\left(\boldsymbol{\mu}_{m,r}^{(t)}, \boldsymbol{\mu}_{n,r}^{(t)}\right)}{\arg\min} \left\{ \langle \mathbf{P}, \mathbf{G} \rangle - \varepsilon \mathcal{H}(\mathbf{P}) \right\}, \tag{22}$$

where $\mathcal{H}(\mathbf{P}) = -\sum_{a,b} P_{ab} \log P_{ab}$ denotes the entropy of the transport plan, and $\varepsilon > 0$ controls the regularization strength. The feasible set $\Pi(\boldsymbol{\mu}, \boldsymbol{\nu})$ contains nonnegative transport plans satisfying $\mathbf{P}\mathbf{1} = \boldsymbol{\mu}$ and $\mathbf{P}^\top \mathbf{1} = \boldsymbol{\nu}$. The resulting transport cost $d_{m,n,r}^{(t)} = \langle \mathbf{P}_\varepsilon^*, \mathbf{G} \rangle$ quantifies the structural discrepancy between the two clients.

A single semantic anchor captures collaboration preferences within a particular context, motivating an overall comparison across the shared anchor collection:

$$d_{m,n}^{(t)} = \frac{1}{R} \sum_{r=1}^{R} d_{m,n,r}^{(t)}. \tag{23}$$

Intuitively, closer collaboration preferences suggest greater potential for knowledge sharing, motivating personalized aggregation guided by structural affinity:

$$\mathcal{K}_{m,n}^{(t)} = \frac{\exp\left(-d_{m,n}^{(t)} / \sigma^{(t)}\right)}{\sum_{k=1}^{M} \exp\left(-d_{m,k}^{(t)} / \sigma^{(t)}\right)}, \tag{24}$$

where $\sigma^{(t)} > 0$ controls the concentration of aggregation weights, with smaller values placing greater emphasis on clients with closer collaboration preferences. Structural affinity thus guides the sharing of locally learned collaboration knowledge, yielding personalized topology-generator parameters for the next communication round:

$$\Theta_m^{t+1} = \sum_{n=1}^{M} \mathcal{K}_{m,n}^{(t)} \Theta_n^{t,+}. \tag{25}$$

These personalized parameters initialize the next round of local structural self-supervision, allowing the transferred collaboration knowledge to be further adapted to local task distributions.

4.4 Personalized Collaboration Inference

The collaboration knowledge learned over T communication rounds is reused at inference through the fixed personalized generator, which constructs a topology for each question:

$$\mathbf{A}^{\inf}_{m,q} = \Gamma_{\Theta^T_m}(q, \mathcal{A}). \tag{26}$$

Under a limited expert budget, participation should reflect both semantic relevance to the question and structural importance within the learned collaboration topology. This motivates a participation score that balances structural connectivity with question–expert semantic affinity:

$$I_{m,q} = \arg\max_{|I|=B} \sum_{i\in I} \left[\mathcal{N}_{\mathcal{A}}\left(\mathbf{A}^{\inf}_{m,q}\mathbf{1} + \left(\mathbf{A}^{\inf}_{m,q}\right)^{\top}\mathbf{1}\right) + \beta\,\mathcal{N}_{\mathcal{A}}\left(\bar{\mathbf{E}}\bar{\mathbf{e}}_q\right)\right]_i, \tag{27}$$

where $I \subseteq \{1,\dots,N\}$ denotes a candidate expert index set, and B specifies the participation budget. $\mathcal{N}_{\mathcal{A}}$ denotes min–max normalization across candidate agents. The selected experts $\mathcal{A}^{\mathrm{act}}_{m,q} = \{a_i \mid i \in I_{m,q}\}$ collaborate through the induced topology $\mathbf{A}^{\mathrm{act}}_{m,q} = \mathbf{A}^{\inf}_{m,q}[I_{m,q}, I_{m,q}]$, retaining the learned interactions among active participants. The active team reasons over question q through these retained interactions, yielding the final prediction:

$$\hat{y}_{m,q} = \mathcal{F}_{\mathrm{MAS}}\left(q, \mathcal{A}^{\mathrm{act}}_{m,q}, \mathbf{A}^{\mathrm{act}}_{m,q}\right). \tag{28}$$

5 Experiments

5.1 Datasets and Baselines

Datasets We evaluate FedGOTA on six public benchmarks spanning four task domains. MMLU and MMLU-Pro cover general reasoning (Hendrycks et al., 2020; Wang et al., 2024), GSM-Hard and AQuA address mathematical reasoning (Gao et al., 2023; Ling et al., 2017), HumanEval evaluates code generation (Chen et al., 2021), and GPQA focuses on scientific reasoning (Rein et al., 2023). To keep the evaluation budget manageable and consistent, we construct fixed experimental subsets with random seed 42 and use exactly the same training and test instances for all methods. Dataset statistics are summarized in Table 2.

Baselines We compare FedGOTA with fourteen representative baselines: single-agent reasoning methods, including Vanilla, CoT (Wei et al., 2022), ComplexCoT (Fu et al., 2022), SC(CoT-5), and SC(ComplexCoT) (Wang et al., 2022); fixed multi-agent collaboration methods, including Chain, Tree, Complete Graph (Qian et al., 2025), and LLM-Debate (Du et al., 2023); and adaptive graph-based methods, including AgentPrune (Zhang et al., 2025), AgentDropout (Wang et al., 2025), G-Designer (Zhang et al., 2024), ARG-Designer (Li et al., 2026a), and OFA-MAS (Li et al., 2026b).

Table 1: Performance comparison. **Mul.**, **Ada.**, and **Fed.** denote multi-agent collaboration, question-adaptive topology, and federated topology learning, respectively. Adaptive methods report mean±std over three runs. The best and second-best means are bolded and underlined, respectively.

Method	Mul.	Ada.	Fed.	MMLU	GSM-Hard	HumanEval	AQuA	MMLU-Pro	GPQA	Avg.
Vanilla	✗	✗	✗	74.43%	47.95%	82.27%	81.28%	54.34%	39.73%	63.33%
CoT	✗	✗	✗	79.91%	49.77%	85.82%	79.00%	58.45%	40.64%	65.60%
ComplexCoT	✗	✗	✗	83.11%	51.60%	84.40%	81.74%	59.36%	40.18%	66.73%
SC(CoT-5)	✗	✗	✗	82.65%	50.23%	85.82%	<u>83.11%</u>	61.19%	41.10%	67.35%
SC(ComplexCoT)	✗	✗	✗	83.56%	51.14%	85.82%	81.28%	61.64%	40.64%	67.35%
Chain	✓	✗	✗	78.54%	53.42%	85.11%	81.28%	60.73%	39.27%	66.39%
Tree	✓	✗	✗	79.45%	50.23%	86.52%	79.45%	59.82%	39.27%	65.79%
Complete Graph	✓	✗	✗	81.74%	52.51%	87.23%	82.65%	56.62%	38.81%	66.59%
LLM-Debate	✓	✗	✗	83.11%	54.34%	<u>87.94%</u>	81.28%	53.88%	<u>41.55%</u>	67.02%
AgentPrune	✓	✓	✗	83.03 ± 0.8%	52.59 ± 2.4%	86.26 ± 0.4%	82.72 ± 0.8%	53.27 ± 0.3%	38.69 ± 1.5%	66.09%
AgentDropout	✓	✓	✗	**84.32** ± 1.6%	<u>58.45</u> ± 2.4%	86.76 ± 1.1%	<u>83.11</u> ± 0.9%	59.06 ± 2.3%	38.81 ± 2.0%	<u>68.42%</u>
G-Designer	✓	✓	✗	81.59 ± 0.5%	56.01 ± 1.3%	85.34 ± 1.6%	82.95 ± 1.6%	56.32 ± 1.9%	40.94 ± 1.3%	67.19%
ARG-Designer	✓	✓	✗	83.17 ± 1.0%	49.62 ± 1.9%	85.11 ± 1.4%	81.28 ± 1.7%	**63.83** ± 2.3%	39.12 ± 1.4%	67.02%
OFA-MAS	✓	✓	✗	81.73 ± 1.2%	48.10 ± 0.7%	86.53 ± 0.7%	81.13 ± 2.3%	58.14 ± 1.0%	37.29 ± 1.9%	65.49%
FedGOTA	✓	✓	✓	<u>84.02</u> ± 0.5%	**61.34** ± 0.7%	**88.91** ± 1.5%	**85.08** ± 0.3%	<u>62.41</u> ± 1.9%	**44.28** ± 1.6%	**71.01%**

5.2 Main Results

As shown in Table 1, FedGOTA achieves competitive performance across the six reasoning benchmarks and attains the best average performance among the compared methods. These results suggest

that decoupling topology learning from task-specific supervision enables collaboration knowledge to be effectively reused across heterogeneous tasks. Meanwhile, structure-aware personalized aggregation allows each client to selectively incorporate collaboration knowledge from structurally compatible clients, helping preserve task-specific collaboration preferences while benefiting from cross-client knowledge transfer. Overall, the results support the effectiveness of combining reusable structural learning with personalized federated collaboration.

5.3 Token Consumption Analysis

To assess collaboration efficiency, we compare FedGOTA with representative multi-agent baselines on MMLU and AQuA. For a fair comparison, all methods are evaluated under the same active-agent budget of five. As shown in Fig. 3, FedGOTA achieves competitive performance on both benchmarks while maintaining relatively low token consumption. Decoupling topology learning from task-specific supervision unifies structural optimization across clients under a shared graph self-supervised objective. Structural pretraining thus avoids token-consuming LLM task execution, helping balance reasoning performance and token cost.

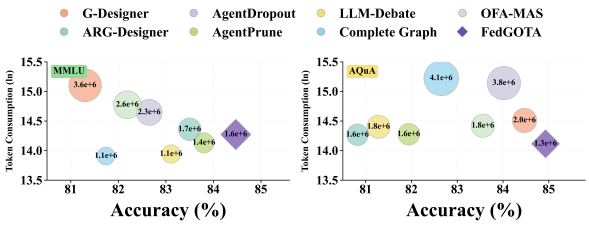


Figure 3: Token Consumption Analysis.

5.4 Aggregation Weight Analysis

To examine whether structural affinity reflects meaningful relationships among clients, we visualize the personalized aggregation weights used during federated training (left) and the corresponding test performance after training (right) in Fig. 4. Consistent with their closely related reasoning domains, MMLU-Pro emerges as the most strongly weighted collaboration partner of MMLU. This agreement between structural affinity and domain relatedness suggests that learned collaboration patterns can provide a meaningful guide for personalized knowledge transfer. Further analyses of aggregation weight scaling and its effects under different collaboration settings are provided in Appendix A.4.

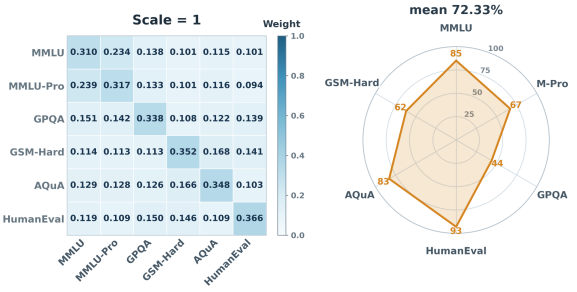


Figure 4: Visualization of personalized aggregation weights and the corresponding performance on the first 100 evaluation instances of each benchmark.

5.5 Ablation Study

To examine the contribution of each structural view to collaboration learning, we conduct ablation studies on the question-aware collaboration view, task-affinity view, and shared structural prior. The gains of the complete formulation in Fig. 5 suggest that collaboration learning benefits from a shared structural reference enriched with question-aware expert interactions and explicit task relevance. Notably, removing the shared structural prior leads to the largest decline in average accuracy, with the degradation on GSM-Hard further highlighting the value of structural guidance for multi-step reasoning. The varying contributions of these views across benchmarks suggest that they complement one another in addressing the diverse collaboration needs of heterogeneous reasoning tasks.

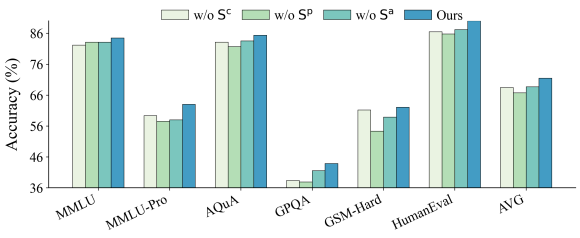


Figure 5: Ablation study.

5.6 Expert Budget Analysis

To assess the value of broader expert participation, we vary the expert budget B across the six benchmarks and examine reasoning accuracy alongside token consumption. As shown in Fig. 6, larger expert budgets increase token consumption without consistently improving accuracy, suggesting that broader participation alone does not ensure more effective collaboration. Broader expert participation may also introduce redundant or irrelevant information, adding noise to the reasoning process and limiting further gains in accuracy. The expert budget should therefore balance the value of additional expertise against the token cost and potential noise associated with broader participation.

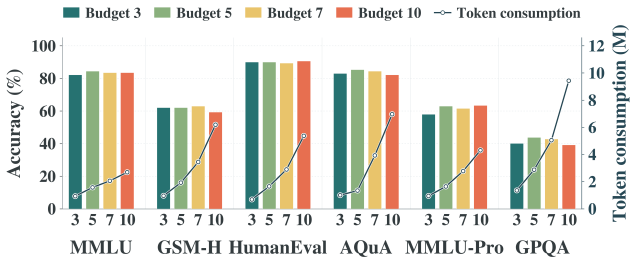


Figure 6: Expert Budget Analysis.

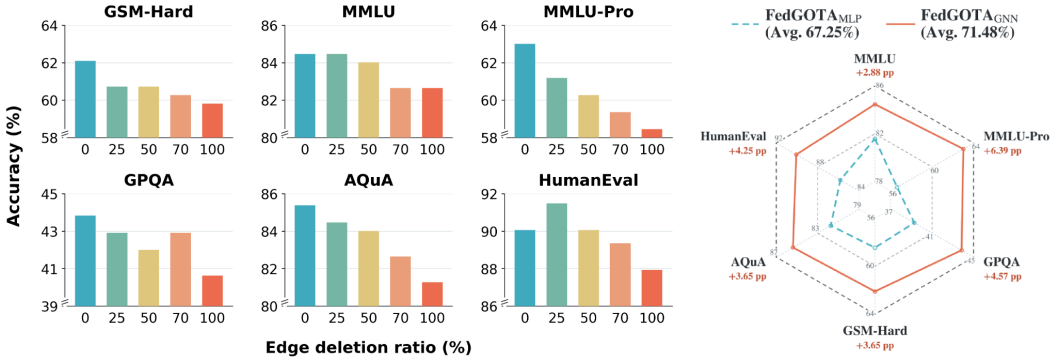


Figure 7: Effectiveness of Learned Graph Structures

5.7 Effectiveness of Learned Graph Structures

To assess the effectiveness of graph structures in collaboration learning, we replace the auxiliary GNN encoder $\mathcal{G}_{\Omega_m}$ with an MLP under the same self-supervised objective. The gains of the GNN-based variant across all six benchmarks in Fig. 7 suggest that the relational information provided by graph structures contributes to more effective multi-agent collaboration. The contribution of the learned communication topology is further examined by progressively removing its edges during inference. Progressively removing edges from the learned communication graphs generally reduces accuracy, with complete removal degrading performance across all six benchmarks. These observations suggest that the learned graphs provide relational guidance during learning and coordinate expert communication during reasoning.

6 Conclusion

In this work, we revisit the limitations of existing adaptive multi-agent collaboration methods and formulate a generalized federated learning objective. Building on this objective, we propose FedGOTA, a federated framework that enables personalized transfer of collaboration knowledge across heterogeneous client tasks. To decouple topology learning from task-specific supervision, a multi-view graph self-supervised objective is introduced to unify structural learning across clients while preserving question-aware expert interactions. Subsequently, graph codebooks constructed under shared semantic contexts are aligned through optimal transport to quantify structural affinity across clients and guide personalized parameter aggregation. Experimental results on six reasoning benchmarks spanning four task domains demonstrate the effectiveness of FedGOTA, highlighting the benefits of structural knowledge reuse for personalized federated collaboration.

AI USE STATEMENT

In this work, we used generative AI tools to polish the language of the manuscript and improve the clarity and consistency of technical descriptions. We have not used generative AI tools to generate references. We have reviewed all AI-assisted revisions to ensure that they preserve the intended meaning and accurately describe the methods and experimental findings. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*, 2022.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models. In *International conference on machine learning*, pp. 10764–10799. PMLR, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Steven Yau, Zijuan Lin, Liyang Zhou, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, volume 2024, pp. 23247–23275, 2024.

Yubin Kim, Chanwoo Park, Hyewon Jeong, Yik S Chan, Xuhai Xu, Daniel McDuff, Hyeonhoon Lee, Marzyeh Ghassemi, Cynthia Breazeal, and Hae W Park. Mdagents: An adaptive collaboration of llms for medical decision-making. *Advances in Neural Information Processing Systems*, 37: 79410–79452, 2024.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in neural information processing systems*, 36:51991–52008, 2023.

Shiyuan Li, Yixin Liu, Qingsong Wen, Chengqi Zhang, and Shirui Pan. Assemble your crew: Automatic multi-agent communication topology design via autoregressive graph generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23142–23150, 2026a.

Shiyuan Li, Yixin Liu, Yu Zheng, Mei Li, Quoc Viet Hung Nguyen, and Shirui Pan. Ofa-mas: One-for-all multi-agent system topology design based on mixture-of-experts graph generative models. In *Proceedings of the ACM Web Conference 2026*, pp. 1333–1344, 2026b.

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. Improving multi-agent debate with sparse communication topology. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7281–7294, 2024.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 17889–17904, 2024.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 158–167, 2017.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. Dynamic llm-agent network: An llm-agent collaboration framework with agent team optimization. *arXiv preprint arXiv:2310.02170*, 2023.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 15174–15186, 2024.

Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, et al. Scaling large language model-based multi-agent collaboration. In *International Conference on Learning Representations*, volume 2025, pp. 41488–41505, 2025.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Kyle Swanson, Wesley Wu, Nash L Bulaong, John E Pak, and James Zou. The virtual lab of ai agents designs new sars-cov-2 nanobodies. *Nature*, 646(8085):716–723, 2025.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290, 2024.

Zhexuan Wang, Yutong Wang, Xuebo Liu, Liang Ding, Miao Zhang, Jie Liu, and Min Zhang. Agentdropout: Dynamic agent elimination for token-efficient and high-performance llm-based multi-agent collaboration. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24013–24035, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Yeyu Yan, Shuai Zheng, Zhenfeng Zhu, Dong Chen, Wentao Zhang, Yawei Zhao, Kunlun He, et al. Aligning condensed graph via hashing: A new insight for federated graph learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Yanwei Yue, Guibin Zhang, Boyang Liu, Guancheng Wan, Kun Wang, Dawei Cheng, and Yiyan Qi. Masrouter: Learning to route llms for multi-agent systems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15549–15572, 2025.

Sukwon Yun, Jie Peng, Pingzhi Li, Wendong Fan, Jie Chen, James Zou, Guohao Li, and Tianlong Chen. Graph-of-agents: A graph-based framework for multi-agent llm collaboration. *arXiv preprint arXiv:2604.17148*, 2026.

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

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Yu, and Tianlong Chen. Cut the crap: An economical communication pipeline for llm-based multi-agent systems. In *International Conference on Learning Representations*, volume 2025, pp. 75389–75428, 2025.

Xiangkai Zhu, Yeyu Yan, Pengpeng Qiao, Tingrui Pei, Yanchun Li, and Saiqin Long. Fedrahi: Reliability-aware hierarchical collaboration for federated graph foundation models. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 6958–6968, 2026.

A Appendix

A.1 Related Work

Static Reasoning and Collaboration. Static paradigms organize reasoning and collaboration through predefined procedures. At the level of individual reasoning, intermediate-step prompting and the aggregation of sampled reasoning paths provide reusable strategies for solving diverse tasks (Wei et al., 2022; Wang et al., 2022). Extending reasoning to interactions among agents, debate-based approaches refine candidate solutions through mutual critique and revision, allowing a common discussion protocol to be adapted across tasks (Du et al., 2023; Liang et al., 2024). The organization of these exchanges is further captured by predefined communication topologies, including sparse connections, chains, trees, and randomly constructed graphs, which offer reusable patterns for coordinating information flow (Li et al., 2024; Qian et al., 2025). Across these approaches, reuse primarily concerns prompting procedures and interaction protocols, while the underlying organization remains largely independent of the input question.

Adaptive Multi-Agent Collaboration. Adaptive paradigms make expert participation and communication structures responsive to task requirements. Expert-selection approaches draw on capability descriptions, contribution estimates, or learned routing to assemble suitable teams from a shared candidate pool, allowing the selection mechanism to serve different questions and tasks (Liu et al., 2023; Yun et al., 2026; Yue et al., 2025). The efficiency of communication among these experts further motivates structural pruning, where redundant agents or links are removed to obtain sparse collaboration patterns that can be reused across queries (Zhang et al., 2025; Wang et al., 2025). Topology-generation approaches introduce greater flexibility by learning mappings from questions to interaction graphs, enabling a shared generator to organize collaboration for different inputs (Zhang et al., 2024; Li et al., 2026a). Training such generators across multiple domains further allows collaboration patterns to be shared among tasks (Li et al., 2026b). Despite these advances, many learned routers and topology generators still rely on task-specific rewards or query–topology supervision, leaving the coordination of heterogeneous local learning objectives an open issue in federated settings.

A.2 Datasets

To maintain a consistent evaluation budget for multi-agent reasoning, we construct fixed experimental subsets from the original benchmarks. For each benchmark except HumanEval, we use random seed 42 to sample 20 instances for local training and 219 instances for evaluation. For HumanEval, we follow the data processing protocol adopted by G-Designer (Zhang et al., 2024) and use its 161-instance experimental pool, from which 20 instances are used for local training and the remaining 141 instances for evaluation. The sampled splits are fixed before experimentation, and all compared methods use exactly the same training and test instances. This controlled subset protocol limits the substantially higher token and computational cost of multi-agent reasoning, where each instance may require multiple LLM calls and inter-agent communication steps. The original evaluation-pool sizes and the subsets used in our experiments are summarized in Table 2.

Table 2: Dataset statistics and subsets used in the main experiments. "Test/Pool" denotes either the official test split or the complete question pool when no separate test split is provided.

			Experimental Pool		Used in Experiments	
Domain	Dataset	Metric	Dev./Val.	Test/Pool	Train	Test
General Reasoning	MMLU	Accuracy	285	14,042	20	219
	MMLU-Pro	Accuracy	70	12,032	20	219
Mathematical Reasoning	GSM-Hard	Accuracy	–	1,319	20	219
	AQuA	Accuracy	254	254	20	219
Code Generation	HumanEval	Pass@1	–	161	20	141
Scientific Reasoning	GPQA	Accuracy	–	448	20	219

A.3 Implementation Details

General Settings. All agents and baselines use GPT-4o-mini as the underlying language model. Federated training comprises two communication rounds, with each client performing 100 local updates per round. For adaptive topology methods, Table 1 reports the mean and standard deviation over three independent runs. Unless otherwise specified, the additional experimental analyses report results from the first run of each configuration. Experiments were conducted on a machine equipped with an Intel Xeon Gold 6230R CPU operating at 2.10 GHz and an NVIDIA A100 GPU with 80 GB of memory, using CUDA 12.8.

Client-Side Implementation. Question and expert representations are obtained from a frozen BERT encoder (Devlin et al., 2019) and mapped into the topology-learning space through a trainable MLP. The graph encoder is implemented as a two-layer GCN (Kipf & Welling, 2016) with a hidden dimension of 64, followed by a projection head for graph self-supervised learning. The shared structural prior is instantiated as a random directed acyclic graph and kept fixed throughout federated training, with its fusion weight set to $\alpha = 0.1$. Local topology generators are optimized using Adam with a learning rate of 1×10^{-2}. Unless otherwise specified, inference uses an active-agent budget of $B = 5$ and a semantic selection weight of $\beta = 1.5$, with the selected experts collaborating through the learned topology.

Server-Side Implementation. Graph codebooks are constructed from 20 shared semantic anchors based on manually defined questions, each comprising two differently worded versions of the same question. The server uses entropy-regularized OT with $\varepsilon = 0.05$ and sets the aggregation temperature to the median inter-client graph distance in each round.

A.4 Effect of Aggregation Weight Scaling

To examine the balance between local model retention and cross-client knowledge sharing, we rescale the learned client affinity matrix. Uniform weights yield equal aggregation across clients, whereas an identity matrix eliminates cross-client aggregation and recovers independent local learning. Average accuracy peaks at an intermediate scale in Fig. 8, highlighting the value of balancing local preferences with selective knowledge sharing across clients. Uniform aggregation overlooks client-specific collaboration preferences, whereas independent local learning forgoes potentially transferable knowledge from other clients. FedGOTA uses structural affinity to tailor cross-client knowledge sharing to the collaboration preferences of each client.

A.5 Effect of Prior Topology

To examine whether the gains of FedGOTA depend on a particular structural prior, we compare FedGOTA with direct execution of different prior topologies. Both settings use the same candidate expert pool, while their active expert sets may differ because FedGOTA performs question-aware expert selection. As shown in Fig. 9, FedGOTA generally improves reasoning performance over direct execution of the corresponding priors, supporting the effectiveness of adaptive collaboration

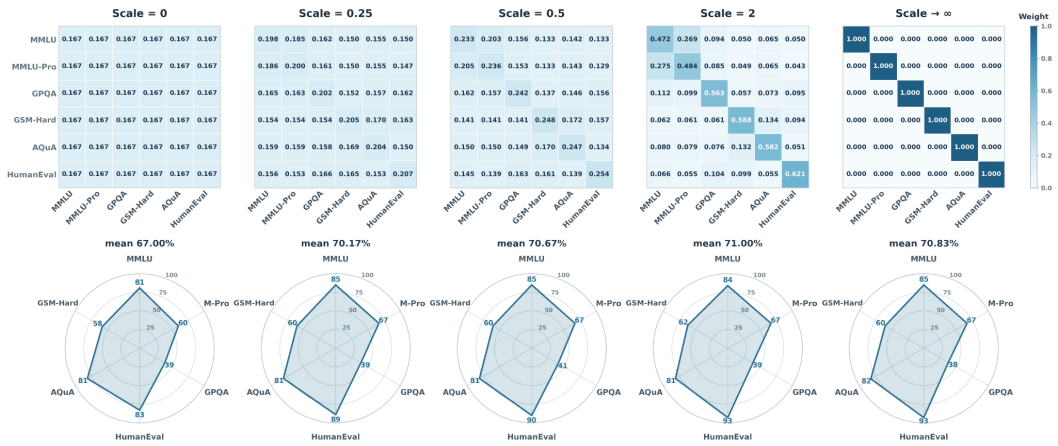


Figure 8: Effect of Aggregation Weight Scaling

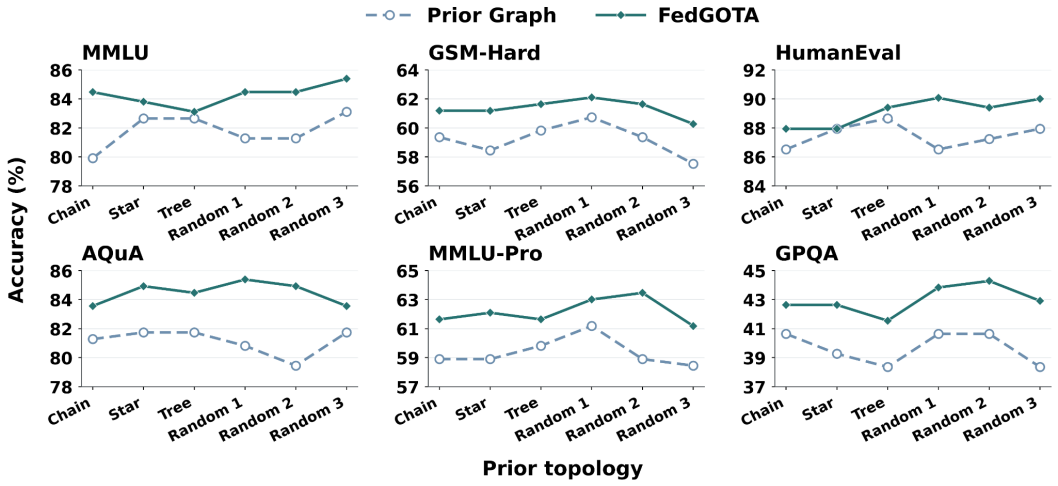


Figure 9: Effect of Prior Topology

under diverse structural guidance. Competitive performance with random priors further suggests that effective structural guidance need not rely on manually designed collaboration patterns.

A.6 Parameter Analysis

The influence of structural guidance and semantic relevance is further examined by varying the prior fusion weight α and semantic selection weight β on MMLU and AQuA. As shown in Fig. 10, the two benchmarks favor different parameter combinations, suggesting that the value of prior guidance and semantic relevance varies with the reasoning demands of each task. Neither stronger prior guidance nor greater semantic emphasis consistently improves accuracy, suggesting that these signals should complement the learned collaboration structure without dominating it. In particular, the gains from incorporating semantic relevance suggest that structural prominence alone is insufficient

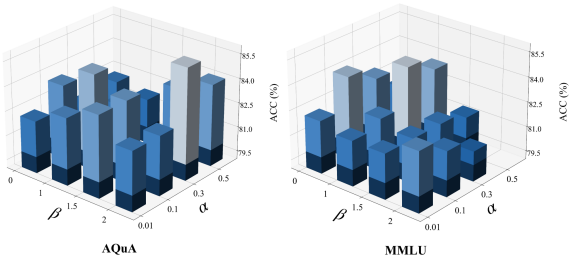


Figure 10: Parameter analysis.

to determine which experts are best suited to a question. These findings favor a balance that preserves shared structural guidance while allowing expert interactions and participation to adapt to the needs of individual questions.

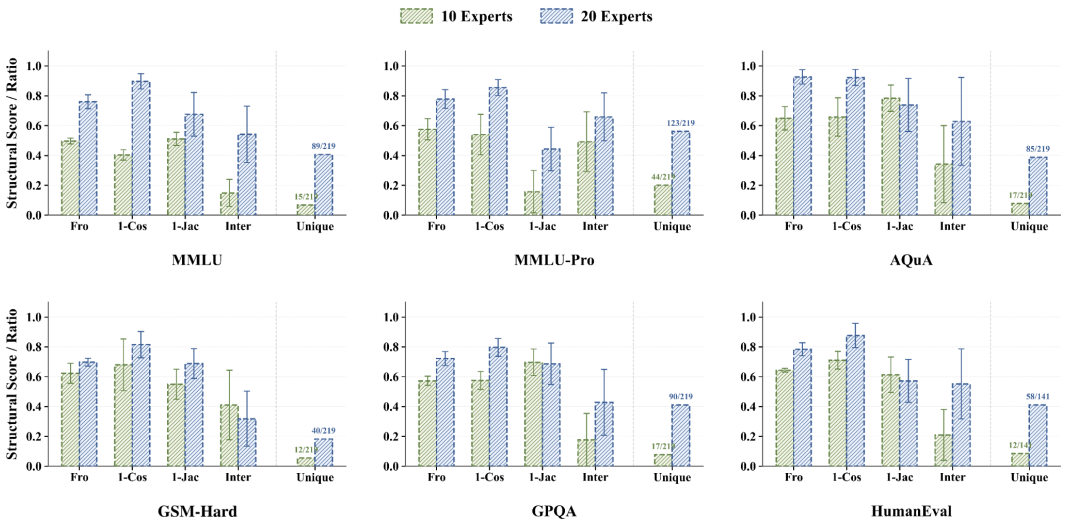


Figure 11: Structural adaptation and diversity. Fro, 1-Cos, and 1-Jac denote the Frobenius distance, cosine dissimilarity, and Jaccard dissimilarity between the learned graphs and the shared prior, respectively. Inter measures structural variation across questions, and Unique denotes the proportion of distinct generated topologies.

A.7 Structural Adaptation and Diversity

To examine whether FedGOTA learns question-aware collaboration patterns from the shared prior, we analyze how the generated graphs differ from this reference and vary across questions. As shown in Fig. 11, the learned graphs deviate from the shared prior across all six benchmarks while exhibiting variation across questions. These observations suggest that the shared prior provides a common structural reference while preserving the flexibility to learn question-aware collaboration patterns. Although a shared prior guides topology learning, the learned graphs differ substantially from this reference, with greater deviations observed as the number of experts increases. This trend highlights the structural adaptability of FedGOTA as the expert population expands. With more experts available, FedGOTA learns a richer variety of question-aware collaboration patterns, further illustrating its flexibility under shared structural guidance. These structural observations complement the earlier performance comparisons, supporting shared priors as reusable guidance for flexible collaboration learning across heterogeneous client tasks.

A.8 Case Study

To illustrate how learned collaboration patterns guide reasoning, we compare the expert responses of FedGOTA and OFA-MAS on an MMLU question. As shown in Fig. 12, FedGOTA brings together factual analysis, independent verification, and evidence synthesis, with complementary expert contributions leading to the correct answer. By comparison, OFA-MAS involves experts whose coding-oriented roles are less relevant to the question, while an incorrect judgment is carried forward in subsequent responses. This case illustrates how learned collaboration structures can bring relevant expertise into a coherent process of verification and synthesis, supporting the final decision with complementary evidence.

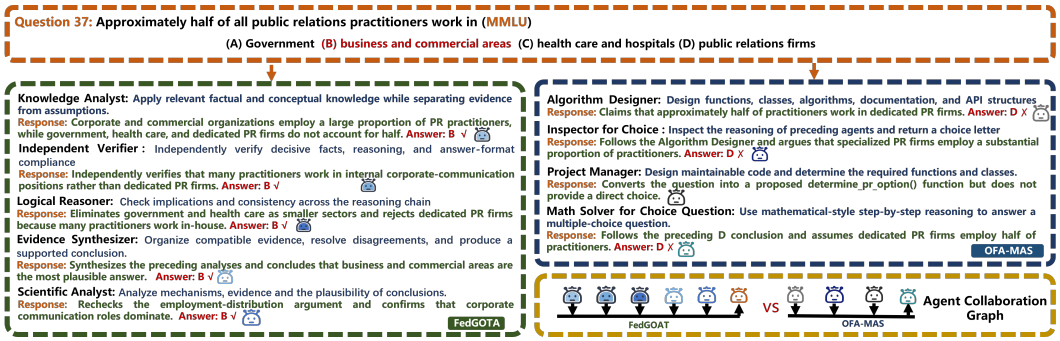


Figure 12: Case study on MMLU

A.9 Computational Complexity of OT Alignment

For each semantic anchor, structural alignment compares two probability distributions supported on $E = N(N-1)$ directed interactions. Under a dense Newton formulation, the resulting OT problem admits a computational complexity of $\mathcal{O}(IE^3)$, where I denotes the number of iterations. Since alignment is performed over R anchors and $\binom{M}{2}$ distinct client pairs, its total time complexity across T communication rounds is

$$\mathcal{O}\left(TR\binom{M}{2}IE^3\right) = \mathcal{O}\left(TRM^2IE^3\right), \tag{29}$$

with a space complexity of $\mathcal{O}(E^2)$ per OT problem. Codebook construction additionally requires MRV topology-generator evaluations per round. As paraphrase-specific responses are averaged before alignment, the transport dimension is independent of V. With the candidate pool fixed at $N = 10$, alignment in our experiments entails 90×90 transport problems across $R = 20$ semantic anchors, each represented by $V = 2$ paraphrases.

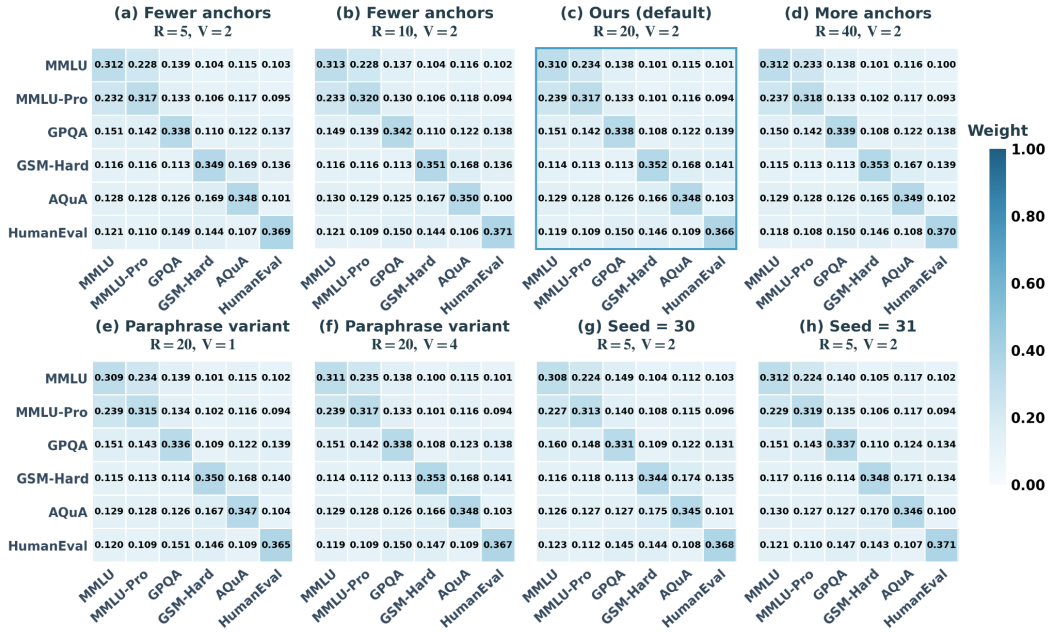


Figure 13: Final-round personalized aggregation weights under different probe configurations and random seeds. Rows and columns correspond to target and source clients, respectively. R and V denote the number of semantic anchors and paraphrases per anchor, respectively.

A.10 Sensitivity to Semantic Probes

To examine the influence of semantic probe construction, we vary the number of anchors, the number of paraphrases per anchor, and the random seed. As shown in Fig. 13, the aggregation weights exhibit similar patterns across these configurations, indicating that the structural relationships among clients are consistently captured by different probe sets.

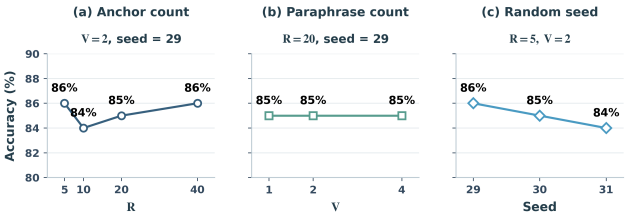


Figure 14: Performance on MMLU under different probe settings.

The corresponding MMLU results in Fig. 14 show relatively stable performance as the probe configuration varies. Increasing the number of anchors or paraphrases does not consistently improve accuracy, suggesting that a compact probe set can provide a useful semantic reference for personalized aggregation.