FEDERATED MULTI-AGENT ORCHESTRATION VIA CONSENSUS-GUIDED INTERACTION DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

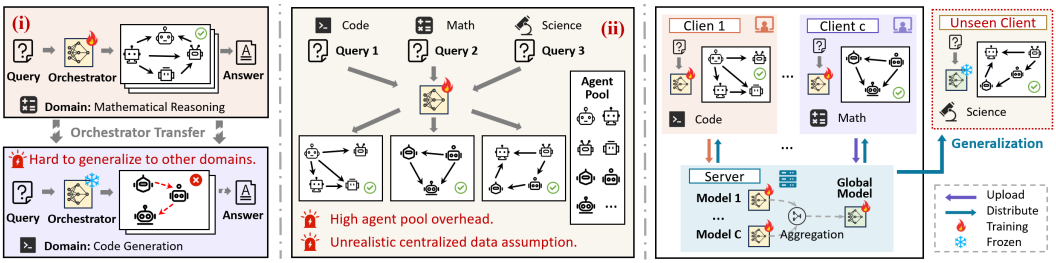
In large language model (LLM)-based multi-agent systems (MAS), the orchestration of inter-agent communication plays a crucial role in improving reasoning quality and collaboration efficiency. However, existing methods typically either learn a dedicated orchestrator for each domain or train a single orchestrator on centrally pooled multi-domain data. These paradigms limit knowledge integration across heterogeneous clients with siloed data, hindering both the learning of a unified orchestrator that accommodates diverse agent configurations and generalization to unseen clients. To address these limitations, we propose **FedGLV**, a federated MAS collaboration framework that learns transferable interaction dynamics for heterogeneous multi-agent orchestration. Specifically, to characterize heterogeneous inter-agent interactions across clients within a common dynamical formulation, we introduce an amortized relational dynamics model inspired by generalized Lotka–Volterra dynamics. Conditioned on queries and agent roles, this model couples latent relational-state evolution with adaptive topology generation, supporting varying numbers of agents and communication steps. At the server, data-free structural consensus distillation is introduced to integrate orchestration knowledge across heterogeneous clients. Concretely, guided by leave-one-out (LOO) valuation, FedGLV aggregates client soft topologies obtained using the server-generated probe bank into a dynamic structural consensus. This consensus provides an explicit structural supervision signal for the global orchestrator to learn transferable interaction patterns. Experiments on six source datasets for federated training and three unseen-client benchmarks demonstrate the effectiveness and generalizability of FedGLV.

1 INTRODUCTION

With the rapid development of large language models (LLMs), LLM-based multi-agent systems (MAS) have emerged as a promising paradigm for tackling complex tasks across diverse scenarios, such as software development, medical decision-making, and scientific discovery (Hong et al., 2024; Du et al., 2024; Qian et al., 2025). These systems assign specialized roles to agents and endow them with complementary capabilities to decompose complex reasoning into coordinated subprocesses. Therefore, the effectiveness of multi-agent collaboration depends not only on individual-agent capability but also on how agents are organized and interact (Zhuge et al., 2024; Du et al., 2024). In particular, the underlying communication topology determines how information is propagated, aggregated, and refined among agents, thereby directly affecting the reasoning quality and efficiency of the overall system (Zhang et al., 2025b; Li et al., 2026b).

To accommodate the diverse collaboration demands arising in LLM-based MAS, recent studies have moved from fixed communication topologies toward adaptive and learnable topologies (Zhuge et al., 2024; Li et al., 2026a). These methods, referred to as adaptive graph-based MAS, typically learn a graph learner or topology generator as an orchestrator to dynamically configure inter-agent interactions. For example, G-Designer (Zhang et al., 2025b) and ARG-Designer (Li et al., 2026a) construct adaptive communication topologies by jointly optimizing agent organization and interaction links. AgentPrune (Zhang et al., 2025a) and AgentDropout (Wang et al., 2025b) further improve collaboration efficiency by eliminating redundant interactions or agents. Moving beyond domain-specific orchestrators, OFA-MAS (Li et al., 2026b) advances this paradigm toward a unified orchestrator across multiple domains.

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.



(a) Limitations of Conventional MAS Methods. (b) A Naive Federated MAS Paradigm.

Figure 1: Motivation and illustration of the federated MAS paradigm.

Despite these advances, existing methods still face practical limitations when deployed across diverse domains, as shown in Fig. 1 (a). Most methods (Zhang et al., 2025b;a; Li et al., 2026a) learn domain-specific orchestrators from individual task distributions, making the learned collaboration patterns difficult to transfer to other domains. Unified multi-domain methods such as OFA-MAS (Li et al., 2026b) rely on centralized access to heterogeneous task data and a large shared agent pool, limiting their practicality in real-world deployments. More importantly, in high-stakes applications such as healthcare and finance (Rieke et al., 2020; Cheng et al., 2021), data are naturally distributed across users or institutions and may not be centrally shared due to privacy and security constraints.

Federated learning (FL) (McMahan et al., 2017; Gu et al., 2026; Yan et al., 2026) enables collaborative model training across clients by exchanging locally learned parameters without centralizing their raw data. This naturally motivates its extension to distributed multi-agent collaboration, as illustrated in Fig. 1 (b). However, learning such a unified orchestrator through federated collaboration across multiple domains is nontrivial due to two fundamental challenges. **(i) Cross-Client Collaboration Heterogeneity.** Due to heterogeneous local resources and application demands, different clients may operate in distinct domains and employ different agent roles, agent populations, and communication steps (Diao et al., 2021; Chen et al., 2024; Zhang et al., 2025b). However, under privacy constraints, these client-specific configurations cannot be shared with the server, posing a significant challenge to learning a global orchestrator. **(ii) Limited Cross-Client Generalization.** At the server, simply aggregating local orchestrators in the parameter space does not necessarily preserve their topology-generation behaviors, since parameter similarity and topology-generation behavior are not inherently aligned (Li et al., 2023; Wan et al., 2026). Moreover, the aggregated model can become tightly coupled to the task distributions of participating clients and fail to capture transferable interaction patterns that generalize to unseen clients and agent configurations (Ma et al., 2024).

To jointly address these challenges, we propose **FedGLV**, a federated framework for learning a unified multi-agent orchestrator across heterogeneous clients. Our key insight is to represent heterogeneous inter-agent interactions within a common dynamical formulation, with their collaboration patterns instantiated through query- and role-conditioned dynamics. Inspired by generalized Lotka–Volterra (GLV) dynamics (Bunin, 2017; Sidhom & Galla, 2020), we formulate multi-agent orchestration as coupled relational-state evolution and topology generation. From this perspective, relational states shape communication topologies, while the resulting interactions drive subsequent state evolution. By conditioning dynamical initialization on query and role semantics, this formulation accommodates configuration-specific collaboration within a shared parameterization, enabling orchestration across heterogeneous configurations. To further consolidate the orchestration knowledge acquired by heterogeneous clients, we propose data-free structural consensus distillation, which establishes a common behavioral reference for local orchestrators through a shared answer-free probe bank. Subsequently, FedGLV performs leave-one-out (LOO) valuation to assess each client's contribution to relational-state agreement and guide the construction of dynamic structural consensus. The resulting consensus provides explicit structural supervision for global orchestration and is distilled into the global orchestrator to enhance generalization across domains and agent configurations.

Our contributions are outlined as follows:

- To the best of our knowledge, we are the first to study federated learning of a unified MAS orchestrator across clients that differ in both task domains and agent configurations. To this end, we develop a federated MAS collaboration framework **FedGLV**, enabling heterogeneous clients to jointly capture transferable MAS interaction patterns without centralizing their private data.

- To formalize inter-agent interactions, we develop a GLV-inspired relational dynamics model that couples relational state evolution with adaptive topology generation. By conditioning dynamical initialization on query and role semantics, it accommodates diverse agent roles, populations, and communication steps within a shared learnable parameterization.
- We propose data-free structural consensus distillation to consolidate the orchestration knowledge acquired by heterogeneous clients. The shared answer-free probe bank and LOO valuation of relational state agreement guide the construction of dynamic structural consensus, providing explicit structural supervision for distilling a global orchestrator to enhance cross-domain generalization.
- Extensive experiments across six benchmarks, together with unseen-client transfer evaluations, demonstrate the effectiveness and generalizability of FedGLV. Systematic ablations and analyses further support component effectiveness, cross-domain transferability, and adaptability to heterogeneous agent configurations.

Related Work. Our work is primarily related to graph-based MAS (Zhang et al., 2025b; Li et al., 2026b) and cross-domain knowledge integration via federated learning (Zhu et al., 2025b; 2026). These studies advance adaptive topology generation and distributed knowledge sharing, providing foundations for transferable MAS collaboration. Building on these directions, FedGLV investigates unified orchestration under joint heterogeneity in client domains and agent configurations. A detailed review of these research directions and their connections to our work is provided in Appendix A.1.

2 PRELIMINARIES

Formulating MAS as Graphs. Given a query q, an MAS with N agents $\mathcal{V} = \{v_i\}_{i=1}^{N}$ is represented over S communication steps by a sequence of directed communication graphs $\{\mathcal{G}^{q,s}\}_{s=0}^{S-1}$, with $\mathcal{G}^{q,s} = (\mathcal{V}, \mathcal{E}^{q,s}, \mathbf{A}^{q,s})$. Each agent v_i is associated with a functional role ρ_i, and $\mathcal{R} = \{\rho_i\}_{i=1}^{N}$ denotes the role set. The adjacency matrix $\mathbf{A}^{q,s} \in \{0,1\}^{N\times N}$ specifies the communication topology at step s, where $[\mathbf{A}^{q,s}]_{ji}$ denotes the communication indicator from v_j to v_i. An edge (v_j, v_i) belongs to $\mathcal{E}^{q,s}$ if and only if $[\mathbf{A}^{q,s}]_{ji} = 1$. The resulting MAS prediction is written as $\hat{y}^q = \mathrm{Exec}\left(q, \mathcal{R}, \{\mathbf{A}^{q,s}\}_{s=0}^{S-1}\right)$, where $\mathrm{Exec}(\cdot)$ denotes the multi-step collaborative reasoning process.

Graph-based Multi-Agent Orchestration. Given a query q and a role set $\mathcal{R}$, graph-based methods typically encode their semantics into a shared representation space and parameterize an orchestrator Φ_Θ to generate query-dependent communication topologies, where Θ denotes its learnable parameters (Zhang et al., 2025b; Li et al., 2026b):

$$\mathbf{e}^q = \mathcal{T}(q), \qquad \mathbf{Z} = [\mathcal{T}(\rho_1), \ldots, \mathcal{T}(\rho_N)]^\top, \qquad \{\mathbf{A}^{q,s}\}_{s=0}^{S-1} = \Phi_\Theta(\mathbf{Z}, \mathbf{e}^q; S), \tag{1}$$

where $\mathcal{T}(\cdot)$ is a frozen semantic encoder, $\mathbf{e}^q \in \mathbb{R}^d$ and $\mathbf{Z} \in \mathbb{R}^{N\times d}$ denote the query embedding and role representation matrix, respectively, and d is the semantic embedding dimension.

Federated Multi-Agent Orchestration. Consider C clients $\mathcal{C} = \{1, \ldots, C\}$, where client c holds a private task distribution $\mathcal{D}_c$ and an agent configuration $\Gamma_c = (\mathcal{R}_c, N_c, S_c)$, with $\mathcal{R}_c = \{\rho_{c,i}\}_{i=1}^{N_c}$. Let $\mathbf{Z}_c = [\mathcal{T}(\rho_{c,1}), \ldots, \mathcal{T}(\rho_{c,N_c})]^\top$ denote the role representation matrix. The local objective is

$$\mathcal{F}_c(\Theta) = \mathbb{E}_{(q,y)\sim\mathcal{D}_c}\left[\mathcal{L}_{\mathrm{task}}\left(\mathrm{Exec}(q, \mathcal{R}_c, \Phi_\Theta(\mathbf{Z}_c, \mathbf{e}^q; S_c)), y\right)\right], \tag{2}$$

where y denotes the ground-truth answer and $\mathcal{L}_{\mathrm{task}}$ is the task-level prediction loss. Federated MAS orchestration aims to learn a global orchestrator by minimizing

$$\Theta_g^* = \arg\min_{\Theta} \sum_{c=1}^{C} \xi_c \cdot \mathcal{F}_c(\Theta), \tag{3}$$

where $\xi_c \geq 0$ denotes the weight assigned to client c in the global objective, with $\sum_{c=1}^{C} \xi_c = 1$. In this setting, private queries, labels, agent configurations, and the collaboration topologies generated from these queries remain local, while only orchestrator parameters are exchanged with the server.

Key Observations. The above formulations reveal two key requirements for federated MAS orchestration under client heterogeneity. (i) The graph orchestrator must accommodate heterogeneous client configurations Γ_c, while maintaining a shared parameterization that can be federated across clients. (ii) Since parameter aggregation may not explicitly preserve topology-generation behaviors, the server must identify and consolidate structural consensus across local orchestrators, so that the resulting global orchestrator Φ_{Θ_g} captures communication patterns shared across clients.

3 METHOD

The framework of FedGLV is illustrated in Fig. 2. At each client, *Semantic Amortization for Dynamic MAS Orchestration* instantiates shared collaboration dynamics from query and role semantics, coupling relational state evolution with adaptive topology generation. At the server, *Data-Free Structural Consensus Distillation* consolidates these local orchestration behaviors into a global orchestrator through structural supervision on shared answer-free probes[1].

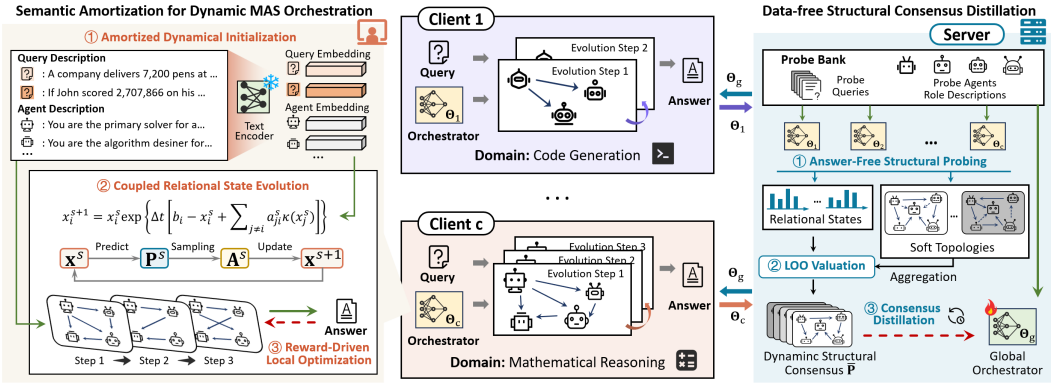


Figure 2: The framework of FedGLV.

3.1 SEMANTIC AMORTIZATION FOR DYNAMIC MAS ORCHESTRATION

Client heterogeneity arises not only from task distributions but also from agent roles, population sizes, and communication steps, making a configuration-specific fixed topology unsuitable for federated MAS orchestration. Generalized Lotka–Volterra (GLV) dynamics (Bunin, 2017; Sidhom & Galla, 2020) provide a promising starting point by expressing each entity's dynamical evolution through intrinsic growth and pairwise influences:

$$\frac{dx_i(t)}{dt} = x_i(t)(b_i + \sum_{j \neq i} \epsilon_{ji} x_j(t)), \tag{4}$$

where $x_i(t)$ denotes the state of entity i, b_i is its intrinsic growth rate, and ϵ_{ji} characterizes the pairwise influence of entity j on entity i.

Evidently, GLV provides a unified dynamical framework for modeling ecological communities across diverse environments, such as oceans, deserts, and forests. Within this framework, intrinsic growth and pairwise interactions follow a common mathematical form, while system-specific coefficients capture distinct ecological dynamics. Motivated by this interaction form, FedGLV adopts GLV-inspired relational dynamics with self-regulation and nonnegative saturating interactions. Shared semantic mappings and a pairwise edge predictor instantiate these dynamics for different queries and agent configurations, providing a structural inductive bias for step-dependent topology generation.

Amortized Dynamical Initialization. Classical GLV assumes a fixed set of coefficients for each interacting system, which is restrictive for MASs whose queries and agent roles vary across clients. Therefore, to instantiate dynamics for different queries and agent configurations, FedGLV predicts initial states and intrinsic growth rates through shared semantic mappings.

For client c's i-th agent, let $\mathbf{z}_{c,i} = [\mathbf{Z}_c]_{i,:}^\top$ denote its role embedding. We fuse it with the query representation to obtain $\mathbf{h}_{c,i}^q = f_h([\mathbf{z}_{c,i} \| \mathbf{e}^q]) \in \mathbb{R}^{d_h}$, where $\|$ denotes concatenation and d_h is the hidden dimension. Two separate heads predict the initial relational state and intrinsic growth rate:

$$x_{c,i}^{q,0} = \mathrm{Softplus}(f_x(\mathbf{h}_{c,i}^q)), \qquad b_{c,i}^q = \mathrm{Softplus}(f_b(\mathbf{h}_{c,i}^q)), \tag{5}$$

where f_h is an MLP and $f_x, f_b : \mathbb{R}^{d_h} \to \mathbb{R}$ are independent scalar-output heads. Here, $x_{c,i}^{q,s} > 0$ denotes the latent relational state of agent v_i at communication step s, while $b_{c,i}^q > 0$ represents its task-conditioned intrinsic growth rate. $\mathrm{Softplus}(\cdot)$ enforces the positivity of both quantities.

[1]For notational simplicity, we omit federated round superscripts in the following derivations.

Coupled Relational State Evolution. Starting from the initialized relational state $\mathbf{x}_c^{q,0}$, FedGLV couples topology generation with relational state evolution across communication steps. At each communication step, FedGLV generates a topology from the collective relational state $\mathbf{x}_c^{q,s} = [x_{c,1}^{q,s}, \ldots, x_{c,N_c}^{q,s}]^\top$ and uses the resulting topology to drive the next state transition. Specifically, the current state modulates each agent's semantic representation as

$$\widetilde{\mathbf{h}}_{c,i}^{q,s} = \mathbf{h}_{c,i}^{q} + \kappa(x_{c,i}^{q,s})\mathbf{U}\mathbf{h}_{c,i}^{q}, \qquad \kappa(x) = \frac{x}{1+x}, \tag{6}$$

where $\mathbf{U} \in \mathbb{R}^{d_h \times d_h}$ is a learnable modulation matrix and $\kappa(x) \in (0,1)$ provides a bounded gate for the state-dependent modulation. The resulting representations determine the soft topology $\mathbf{P}_c^{q,s} = [p_{c,ji}^{q,s}] \in [0,1]^{N_c \times N_c}$ through

$$p_{c,ji}^{q,s} = \sigma\Big(f_e\Big([\widetilde{\mathbf{h}}_{c,j}^{q,s} \| \widetilde{\mathbf{h}}_{c,i}^{q,s}]\Big)\Big), \qquad j \neq i, \tag{7}$$

where $f_e : \mathbb{R}^{2d_h} \to \mathbb{R}$ is a shared pairwise edge predictor and $\sigma(\cdot)$ is the sigmoid function. During training, the executable topology $\mathbf{A}_c^{q,s} = [a_{c,ji}^{q,s}]$ is obtained by conditionally independent sampling, $a_{c,ji}^{q,s} \sim \mathrm{Bernoulli}(p_{c,ji}^{q,s})$ for $j \neq i$, with $a_{c,ii}^{q,s} = 0$.

To instantiate the dynamical formulation over discrete communication steps, FedGLV formulates topology-conditioned relational dynamics with self-regulation and nonnegative saturating interactions. Given the topology $\mathbf{A}_c^{q,s}$, Euler discretization in log-state space yields the relational state update:

$$x_{c,i}^{q,s+1} = x_{c,i}^{q,s} \exp\left\{ \Delta t \left[b_{c,i}^{q} - x_{c,i}^{q,s} + \sum_{j \neq i} a_{c,ji}^{q,s} \kappa\big(x_{c,j}^{q,s}\big) \right] \right\}, \tag{8}$$

where $\Delta t = 1/S_c$ is the client-specific discretization step size, $\kappa(x)$ is the saturating function defined in Eq. (6), and $-x_{c,i}^{q,s}$ provides self-regulation to control state growth. Therefore, relational states and communication topologies co-evolve, i.e., the states condition topology generation, while the resulting topologies in turn govern subsequent state updates.

In the above formulation, the learnable parameters of f_h, f_x, f_b, f_e, and $\mathbf{U}$ are collected in Θ_c. Through shared semantic mappings, FedGLV flexibly instantiates query-conditioned relational dynamics for diverse role compositions and agent populations. Client-specific discretization further allows the relational dynamics to unfold over different numbers of communication steps.

Reward-Driven Local Optimization. Since optimal communication topologies are unavailable as supervision, FedGLV optimizes local orchestrators directly from downstream task feedback. Following G-Designer (Zhang et al., 2025b), each client adopts a small set of labeled task examples for local training. For each $(q, y) \sim \mathcal{D}_c$, the resulting MAS prediction $\hat{y}_c^q$ receives a reward $R_c^q = \mathrm{Score}(\hat{y}_c^q, y) \in [0,1]$, where $\mathrm{Score}(\cdot)$ denotes the task-specific evaluation function and is instantiated as binary task correctness in our experiments. Specifically, the executable topology $\mathbf{A}_c^{q,s}$ at step s is sampled from the Bernoulli policy parameterized by the soft topology $\mathbf{P}_c^{q,s}$. FedGLV optimizes these sequential topology decisions with REINFORCE:

$$\mathcal{L}_{\mathrm{task}}^{c} = -\mathbb{E}_{(q,y)\sim\mathcal{D}_c}[(R_c^q - \bar{R}_c)\frac{1}{S_c}\sum_{s=0}^{S_c-1} \log \pi_{\Theta_c}(\mathbf{A}_c^{q,s} \mid \mathbf{P}_c^{q,s})], \tag{9}$$

where $\bar{R}_c$ is an exponential moving-average reward baseline, and $\pi_{\Theta_c}(\mathbf{A}_c^{q,s} \mid \mathbf{P}_c^{q,s})$ denotes the factorized Bernoulli likelihood over all non-self-loop edges.

To further encourage effective collaboration with fewer communication links, FedGLV imposes a sparsity penalty on the learned soft topologies to discourage redundant interactions:

$$\mathcal{L}_{\mathrm{sparse}}^{c} = \mathbb{E}_{q\sim\mathcal{D}_c}[\frac{1}{S_c}\sum_{s=0}^{S_c-1} \mathrm{Mean}_{i\neq j}([\mathbf{P}_c^{q,s}]_{ji})], \tag{10}$$

where $\mathrm{Mean}_{i\neq j}(\cdot)$ averages over all non-self-loop entries. The resulting local objective is

$$\mathcal{L}_c = \mathcal{L}_{\mathrm{task}}^{c} + \lambda_{\mathrm{s}}\mathcal{L}_{\mathrm{sparse}}^{c}, \tag{11}$$

where λ_{s} controls the trade-off between task-driven collaboration and communication sparsity. Starting from the global orchestrator, each client minimizes Eq. (11) on its private tasks to obtain the locally adapted parameters Θ_c, which are returned to the server for structural consensus distillation.

3.2 DATA-FREE STRUCTURAL CONSENSUS DISTILLATION

After local optimization, the server receives a collection of local orchestrators $\{\Theta_c\}_{c=1}^{C}$, each encoding collaboration patterns learned from its own private data. However, these patterns are expressed through nonlinear topology-generation dynamics, so naively aggregating the orchestrator parameters may not yield a corresponding consensus of their topologies (Li et al., 2023; Wan et al., 2026). Consequently, conventional federated methods based on parameter aggregation may lead to suboptimal collaboration in MAS orchestration. Inspired by data-free distillation (Chen et al., 2019; Zhang et al., 2022), FedGLV introduces a consensus of local communication topologies as explicit supervision for enhancing the global orchestrator's ability to learn transferable interaction patterns.

Answer-Free Structural Probing. To make heterogeneous local orchestrators directly comparable, the server independently establishes a shared probe bank $\mathcal{B} = \{B^m = (q^m, \mathcal{R}^m, S^m)\}_{m=1}^{M}$, where $B^m = (q^m, \mathcal{R}^m, S^m)$ specifies a query, a canonical role set, and a number of communication steps. The probes contain no reference answers or client-specific samples, and their query and role representations are obtained using the frozen encoder $\mathcal{T}(\cdot)$ defined in Section 2.

For each probe B^m, every local orchestrator Θ_c is evaluated under the same query, role configuration, and communication steps. Server-side probing uses predicted edge probabilities in place of sampled edge indicators in Eq. (8), with $\Delta t = 1/S^m$. At communication step s, orchestrator Θ_c produces the relational state vector $\mathbf{x}_c^{m,s} \in \mathbb{R}_+^{N_m}$ and the corresponding state-conditioned soft topology $\mathbf{P}_c^{m,s} = [p_{c,ji}^{m,s}] \in [0,1]^{N_m \times N_m}$, where $N_m = |\mathcal{R}^m|$ is the number of roles in the probe.

By providing the same roles to all local orchestrators, each probe establishes a shared semantic reference for their states and topologies, making the collaboration patterns learned by clients comparable.

LOO Valuation-Guided Structural Consensus. With the probe input fixed, differences in relational states and soft topologies reflect the behavior encoded by local orchestrator parameters. More importantly, the relational states directly modulate agent representations in Eq. (6), from which the soft topologies are generated through Eq. (7). This dependence motivates relational-state agreement as an answer-free proxy for weighting client topologies. Following the marginal contribution principle (Koh & Liang, 2017; Zhu et al., 2025a), FedGLV extends leave-one-out (LOO) valuation to quantify each client's contribution to this agreement and determine its consensus weight. Specifically, for a fixed probe m and communication step s, the state vector is first normalized as $\widetilde{\mathbf{x}}_c^{m,s} = \frac{\mathbf{x}_c^{m,s}}{\|\mathbf{x}_c^{m,s}\|_2}$. For any client subset $\mathcal{S} \subseteq \mathcal{C}$ with $|\mathcal{S}| \geq 2$, we define the subset utility as the average pairwise agreement

$$\mathrm{U}^{m,s}(\mathcal{S}) = \frac{2}{|\mathcal{S}|(|\mathcal{S}|-1)} \sum_{\substack{c<l \\ c,l \in \mathcal{S}}} (\widetilde{\mathbf{x}}_c^{m,s})^\top \widetilde{\mathbf{x}}_l^{m,s}. \tag{12}$$

Then, the value of client c can be defined as

$$v_c^{m,s} = \mathrm{U}^{m,s}(\mathcal{C}) - \mathrm{U}^{m,s}(\mathcal{C} \setminus \{c\}), \tag{13}$$

where $v_c^{m,s}$ measures client c's marginal contribution to relational-state agreement for probe m and communication step s. Building on this, FedGLV treats clients' valuations as weighting signals to guide the combination of local soft communication topologies into a dynamic structural consensus:

$$\bar{\mathbf{P}}^{m,s} = \sum_{c=1}^{C} \omega_c^{m,s} \mathbf{P}_c^{m,s}, \qquad \omega_c^{m,s} = \frac{\exp(v_c^{m,s})}{\sum_{l=1}^{C} \exp(v_l^{m,s})}. \tag{14}$$

For each probe m, FedGLV aggregates local soft topologies across clients separately at each communication step s. This stepwise aggregation yields a dynamic structural consensus $\{\bar{\mathbf{P}}^{m,s}\}_{s=0}^{S^m-1}$ that captures the dynamic evolution of collaboration patterns across the probe's communication steps.

Proposition 1 (Relational-State Agreement under LOO Weighting)**.** *For a fixed probe-step pair, suppress the superscripts (m, s). If $C \geq 3$, the LOO softmax weights satisfy $\frac{\sum_{c<l} \omega_c \omega_l \widetilde{\mathbf{x}}_c^\top \widetilde{\mathbf{x}}_l}{\sum_{c<l} \omega_c \omega_l} \geq \mathrm{U}(\mathcal{C})$. The inequality is strict unless all client LOO values are equal.*

Remark. For fixed relational states, Proposition 1 shows that LOO weighting yields no lower weighted pairwise agreement than uniform weighting. The downstream benefit of LOO weighting for structural distillation is evaluated in Table 3. The proof is provided in Appendix A.2.

Consensus Distillation. After constructing the dynamic structural consensus, FedGLV distills it into the global orchestrator through explicit supervision.

Initialized at $\Theta_{\text{init}} = \frac{1}{C}\sum_{c=1}^{C}\Theta_c$, the global orchestrator generates soft topologies on the same probe bank, producing $\widehat{\mathbf{P}}_{\Theta}^{m,s}$ for each probe m and communication step s. Subsequently, the global parameters are optimized to reproduce the valuation-guided structural consensus:

$$\Theta_g = \arg\min_{\Theta} \frac{1}{M}\sum_{m=1}^{M}\frac{1}{S^m}\sum_{s=0}^{S^m-1}\mathrm{BCE}\left(\bar{\mathbf{P}}^{m,s}, \widehat{\mathbf{P}}_{\Theta}^{m,s}\right), \tag{15}$$

where $\mathrm{BCE}(\cdot,\cdot)$ denotes the mean soft-target binary cross-entropy over non-self-loop edges. By jointly matching the structural consensus across communication steps, FedGLV distills the dynamic topology-generation behaviors of heterogeneous local orchestrators into a global orchestrator for the next federated round and downstream deployment.

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate FedGLV on six datasets: MMLU (Hendrycks et al., 2020), MMLU-Pro (Wang et al., 2024), GSM-Hard (Gao et al., 2023), AQuA (Ling et al., 2017), HumanEval (Chen et al., 2021) and GPQA (Rein et al., 2023). In addition, to assess the generalization of FedGLV to unseen clients, we further evaluate FedGLV on three datasets representing unseen clients, including SAT-Math (Zhong et al., 2024), MedQA (Jin et al., 2021), and LEXam (Fan et al., 2026).

Baselines. In this work, FedGLV is compared with (i) single agent methods, including CoT (Wei et al., 2022) and SC (CoT) (Wang et al., 2023); (ii) fixed multi-agent topologies and LLM-Debate (Du et al., 2024); and (iii) adaptive graph-based methods, including AgentPrune (Zhang et al., 2025a), AgentDropout (Wang et al., 2025b), G-Designer (Zhang et al., 2025b), ARG-Designer (Li et al., 2026a), and OFA-MAS (Li et al., 2026b). To support federated scenarios, we develop federated variants of G-Designer and ARG-Designer with the numbers of agents and communication steps kept consistent across clients, which are denoted as GDesigner$_F$ and ARGDesigner$_F$, respectively.

Evaluation and Implementation. We report the success rate on the test set. In the federated setting, each dataset is treated as an individual client. FedGLV, GDesigner$_F$, and ARGDesigner$_F$ are trained collaboratively across clients. OFA-MAS is evaluated in its centralized setting by jointly using six datasets, while the remaining baselines are executed or trained independently on each client. The best and second-best results are highlighted in bold and underlined, respectively. Details of the experimental setup and FedGLV are provided in Appendices A.3 and A.4, respectively.

Table 1: Performance (%) comparison on six heterogeneous source clients.

Method	MMLU	GSM-Hard	HumanEval	AQuA	MMLU-Pro	GPQA	Avg.
Single Agent (Single Domain)							
Vanilla	76.10±0.70	47.25±0.53	82.71±0.41	79.16±1.40	52.92±0.46	35.92±0.95	62.34
CoT	80.50±1.32	50.53±1.40	84.63±0.41	79.97±1.32	59.82±1.37	37.60±2.16	65.51
SC (CoT)	82.09±0.46	50.23±0.79	85.29±1.48	80.82±1.37	60.19±1.21	40.79±0.70	66.57
Multi-Agent (Single Domain)							
Chain	79.30±0.26	52.51±0.70	84.16±2.69	79.15±1.60	58.30±0.53	39.73±0.91	65.53
Tree	79.76±0.70	52.82±1.15	84.40±2.13	80.52±1.32	57.84±0.95	40.33±0.95	65.95
LLM-Debate	81.58±0.95	53.42±1.21	84.40±1.88	80.82±1.21	57.69±1.47	39.57±0.70	66.25
AgentPrune	83.03±0.81	52.59±2.44	86.26±0.38	82.72±0.81	53.27±0.26	38.69±1.48	66.09
AgentDropout	84.32±1.60	**58.45±2.37**	86.76±1.08	83.11±0.92	59.06±2.34	38.81±1.99	68.42
G-Designer	81.59±0.52	56.01±1.26	85.34±1.64	82.95±1.60	56.32±1.87	40.94±1.29	67.19
ARG-Designer	83.17±1.04	49.62±1.90	85.11±1.42	81.28±1.65	**63.83±2.25**	39.12±1.40	67.02
Multi-Agent (Centralized Multi-Domain)							
OFA-MAS	81.73±1.21	48.10±0.70	86.53±0.71	81.13±2.30	58.14±0.95	37.29±1.85	65.49
Federated Multi-Agent (Distributed Multi-Domain)							
GDesigner$_F$	79.00±1.21	52.05±1.21	84.63±1.48	81.58±1.46	55.10±2.35	39.73±1.65	65.35
ARGDesigner$_F$	82.19±1.99	48.40±0.16	85.11±1.08	82.65±0.70	60.27±0.52	37.75±0.95	66.06
FedGLV	**84.41±0.27**	57.99±0.91	**87.71±0.41**	**83.26±0.70**	60.43±0.95	**41.86±0.70**	**69.28**

4.2 Experimental Results

Performance on Heterogeneous Source Clients. Table 1 reports the performance on the six heterogeneous source clients. FedGLV achieves the best average performance of 69.28%. Across individual clients, FedGLV achieves the best performance on MMLU, HumanEval, AQuA, and GPQA, while remaining competitive on GSM-Hard and MMLU-Pro. Notably, directly extending G-Designer and ARG-Designer with FedAvg decreases their average performance from 67.19% to 65.35% and from 67.02% to 66.06%, respectively, indicating that conventional parameter aggregation may fail to preserve topology-generation knowledge under cross-domain heterogeneity. In contrast, FedGLV consistently integrates orchestration knowledge across heterogeneous clients while maintaining strong performance across diverse domains.

Generalization to Unseen Clients. To evaluate the generalizability of the learned global orchestrator, we consider three unseen clients, SAT-Math, MedQA, and LEXam, spanning mathematical, medical, and legal reasoning. During evaluation, the orchestrator is frozen after training on the six source datasets and directly deployed to these clients without parameter updates. As shown in Table 2, FedGLV achieves the best average performance of 71.80%. It remains competitive with the best baseline on SAT-Math, while achieving 73.95% on MedQA and 46.78% on LEXam, outperforming the second-best methods by 1.96 and 3.92 percentage points, respectively. These results suggest that FedGLV captures transferable interaction patterns that support effective collaboration across diverse tasks and generalize beyond the participating clients.

Table 2: Generalization to unseen clients.

Method	SAT-Math	MedQA	LEXam	Avg.
AgentPrune	92.44±1.68	71.99±0.49	38.94±1.75	67.79
AgentDropout	91.88±2.57	70.31±1.28	38.38±1.28	66.86
G-Designer	94.40±0.49	66.95±0.97	40.06±0.97	67.14
ARG-Designer	94.68±0.49	70.59±0.84	37.25±1.28	67.51
OFA-MAS	89.92±2.91	70.03±2.11	40.34±1.46	66.76
GDesigner$_F$	92.44±2.22	68.91±0.84	39.78±1.28	67.04
ARGDesigner$_F$	**94.96±0.84**	71.15±0.49	42.86±0.84	69.66
FedGLV	94.68±0.97	**73.95±0.84**	**46.78±1.28**	**71.80**

Effect of Heterogeneous Agent Configurations. To evaluate FedGLV's ability to accommodate heterogeneous configurations, the number of agent roles is varied over $\{3, 5, 7\}$ and the number of communication steps S over $\{1, \ldots, 5\}$. Fig. 3 reveals two observations. Notably, increasing the number of roles generally improves performance, particularly with fewer communication steps. At $S = 2$, accuracy increases from 57.53% with three roles to 60.73% with seven roles. This suggests that complementary role perspectives can provide broader reasoning coverage and additional opportunities for verification. Moreover, extending communication beyond a moderate number of communication steps can reduce performance. With seven roles, accuracy peaks at $S = 2$ and declines to 57.99% at $S = 5$. A possible explanation is that prolonged exchanges introduce redundant information and propagate intermediate errors, while repeated exposure to shared responses reduces the diversity of agent judgments. The results demonstrate FedGLV's applicability across varied configurations and suggest that role diversity and communication steps should be considered jointly.

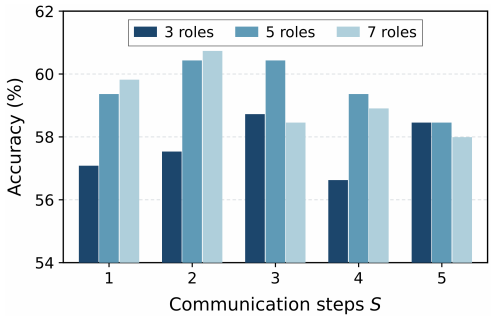


Figure 3: Performance of FedGLV on MMLU-Pro under heterogeneous configurations.

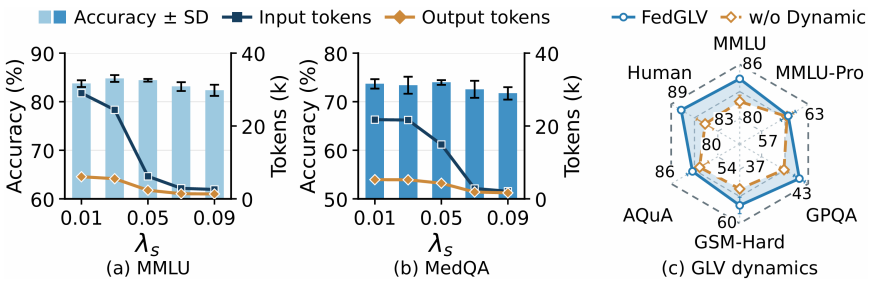


Figure 4: Sparsity regularization analysis (a–b) and relational state evolution ablation (c).

Sparsity Regularization and Relational-State Evolution. Fig. 4 (a) and (b) examine the influence of the sparsity regularization coefficient λ_s. The bars show the accuracy and the two curves report the average input and output token consumption per question. Increasing λ_s reduces both input and output token consumption while largely maintaining accuracy under moderate regularization. Further increasing λ_s degrades accuracy, indicating a trade-off between communication efficiency and task performance. Fig. 4 (c) ablates relational-state evolution by fixing each relational state at its initialized value throughout all communication steps, denoted as *w/o Dynamic*. This variant underperforms FedGLV on all six datasets, with an average accuracy drop of 1.50 percentage points. These results indicate that the performance gain is not solely attributable to semantic initialization. Evolving relational states across communication steps provides additional information for generating step-dependent communication topologies.

Analysis of LOO Consensus Weights. To understand how different clients contribute to collaboration learning, we visualize the average LOO consensus weights over five federated rounds in Fig. 5. The weights exhibit clear client- and round-dependent dynamics in the early stages. For example, AQuA and MMLU receive relatively high weights initially, while the weight of HumanEval increases overall from 13.49% to 17.03%. Notably, as training proceeds, the client weights gradually move toward the uniform reference of $1/6$, indicating more balanced weighting in structural consensus construction.

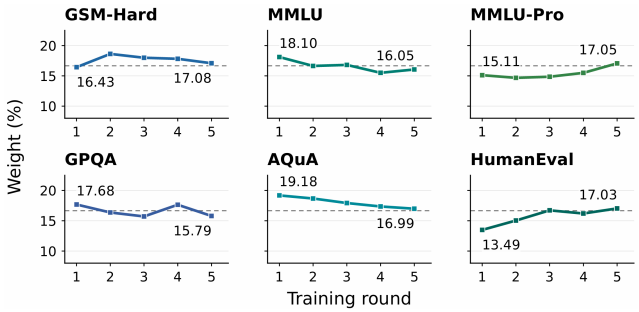


Figure 5: Average LOO consensus weights across federated rounds.

Ablation Study. The contributions of structural consensus distillation and LOO valuation are examined through two variants. The *w/ FedAvg* variant replaces structural consensus distillation with FedAvg parameter aggregation, while *w/o LOO* retains distillation but constructs the structural consensus by uniformly averaging client soft topologies. As shown in Table 3, the FedAvg variant performs worse than FedGLV on all three unseen-client benchmarks, with the largest decline of 4.20 percentage points on LEXam. This suggests that explicit structural supervision is more effective at transferring orchestration behavior to unseen clients. Removing LOO valuation also generally reduces performance, with accuracy dropping by 2.89 and 2.24 points on AQuA and LEXam, respectively. Although the gains vary, these results provide empirical evidence that relational-state agreement can serve as a useful proxy for weighting client topologies during structural consensus construction.

Table 3: Server-side ablations. Shaded rows report generalization to unseen clients.

Dataset	FedGLV	w/ FedAvg	w/o LOO
MMLU	**84.41**±0.27	81.74±1.21	83.56±0.79
GSM-Hard	**57.99**±0.91	56.93±0.70	56.16±1.21
HumanEval	87.71±0.41	84.16±1.64	**87.94**±1.23
AQuA	**83.26**±0.70	82.34±1.15	80.37±0.79
MMLU-Pro	60.43±0.95	**60.88**±0.53	59.36±1.21
GPQA	41.86±0.70	40.94±0.95	**42.01**±0.46
Avg.	**69.28**	67.83	68.23
SAT-Math	94.68±0.97	94.40±1.28	**94.96**±0.84
MedQA	**73.95**±0.84	71.99±0.97	72.83±1.28
LEXam	**46.78**±1.28	42.58±1.75	44.54±1.68
Avg.	**71.80**	69.66	70.78

5 Conclusions

In this work, we propose FedGLV, a federated framework for learning a unified MAS orchestrator across heterogeneous domains and agent configurations. Its amortized GLV-inspired relational dynamics model couples relational-state evolution with adaptive topology generation through query- and role-conditioned initialization, supporting diverse agent roles, populations, and communication steps within a shared parameterization. At the server, data-free structural consensus distillation further consolidates orchestration behaviors through a shared probe bank and LOO valuation, without accessing raw client data or task labels. The resulting dynamic structural consensus provides explicit supervision for capturing cross-client orchestration knowledge and transferable interaction patterns. Experiments demonstrate the effectiveness of FedGLV on participating clients and its generalization to unseen clients.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and assistance with code implementation and debugging. All AI-assisted text and code were reviewed and revised by the authors, and the resulting implementations were tested and verified for correctness. We also independently verified the mathematical derivations, experimental results, and references. We take responsibility for the final content of this work, including text, code, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Guy Bunin. Ecological communities with lotka-volterra dynamics. *Physical Review E*, 95(4):042414, 2017.

Canyu Chen, Kangyu Zhu, Zhaorun Chen, Zhanhui Zhou, Shizhe Diao, Yiping Lu, Tian Li, Manling Li, and Dawn Song. Is decentralized llm agent rl robust to heterogeneity? an asymmetric tale. *arXiv preprint*, 2026. URL `https://fed-agent.github.io/`.

Guangyao Chen, Siwei Dong, Yu Shu, Ge Zhang, Jaward Sesay, Börje Karlsson, Jie Fu, and Yemin Shi. Autoagents: A framework for automatic agent generation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 22–30, 2024.

Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3513–3521, 2019.

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.

Xiang Chen, Yuling Shi, Qizhen Lan, Yuchao Qiu, Min Wang, Xiaodong Gu, and Yanfu Yan. Fed-se: Federated self-evolution for privacy-constrained multi-environment llm agents. *arXiv preprint arXiv:2512.08870*, 2025.

Kewei Cheng, Tao Fan, Yilun Jin, Yang Liu, Tianjian Chen, Dimitrios Papadopoulos, and Qiang Yang. Secureboost: A lossless federated learning framework. *IEEE Intell. Syst.*, 36(6):87–98, 2021.

Enmao Diao, Jie Ding, and Vahid Tarokh. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. In *International Conference on Learning Representations (ICLR)*, 2021.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *International Conference on Machine Learning (ICML)*, volume 235, pp. 11733–11763, 2024.

Yu Fan, Jingwei Ni, Jakob Merane, Yang Tian, Yoan Hermstrüwer, Yinya Huang, Mubashara Akhtar, Etienne Salimbeni, Florian Geering, Oliver Dreyer, et al. Lexam: Benchmarking legal reasoning on 340 law exams. In *International Conference on Learning Representations (ICLR)*, pp. 43451–43489, 2026.

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 (ICML)*, pp. 10764–10799, 2023.

Hanlin Gu, Jiahuan Luo, Yan Kang, Yuan Yao, Gongxi Zhu, Bowen Li, Lixin Fan, and Qiang Yang. Fedadob: Privacy-preserving federated deep learning with adaptive obfuscation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–12, 2026.

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, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. Metagpt: Meta programming for A multi-agent collaborative framework. In *International Conference on Learning Representations (ICLR)*, 2024.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning (ICML)*, pp. 1885–1894, 2017.

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 *the AAAI Conference on Artificial Intelligence (AAAI)*, 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 *the ACM Web Conference (WWW)*, pp. 1333–1344, 2026b.

Xunkai Li, Zhengyu Wu, Wentao Zhang, Yinlin Zhu, Ronghua Li, and Guoren Wang. Fedgta: Topology-aware averaging for federated graph learning. *Proc. VLDB Endow.*, 17(1):41–50, 2023.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 158–167, 2017.

Mengmeng Ma, Tang Li, and Xi Peng. Beyond the federation: Topology-aware federated learning for generalization to unseen clients. In *International Conference on Machine Learning (ICML)*, volume 235, pp. 33794–33810, 2024.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1273–1282, 2017.

Farhad Pourpanah, Mahdiyar Molahasani, Milad Soltany, Michael Greenspan, and Ali Etemad. Federated unsupervised domain generalization using global and local alignment of gradients. In *the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19948–19958, 2025.

Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Scaling large language model-based multi-agent collaboration. In *International Conference on Learning Representations (ICLR)*, 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.

Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletarì, Holger R. Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N. Galtier, Bennett A. Landman, Klaus H. Maier-Hein, Sébastien Ourselin, Micah J. Sheller, Ronald M. Summers, Andrew Trask, Daguang Xu, Maximilian Baust, and M. Jorge Cardoso. The future of digital health with federated learning. *npj Digit. Medicine*, 3, 2020.

Zitong Shi, Guancheng Wan, Wenke Huang, Guibin Zhang, Jiawei Shao, Mang Ye, and Carl Yang. Privacy-enhancing paradigms within federated multi-agent systems. *arXiv preprint arXiv:2503.08175*, 2025.

Laura Sidhom and Tobias Galla. Ecological communities from random generalized lotka-volterra dynamics with nonlinear feedback. *Physical Review E*, 101(3):032101, 2020.

Guancheng Wan, Mo Zhou, Ziyi Wang, Xiaoran Shang, Eric Hanchen Jiang, Guibin Zhang, Jinhe Bi, Yunpu Ma, Zaixi Zhang, Ke Liang, et al. Dawn: Distributed llm multi-agent workflow synthesis. In *the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26099–26106, 2026.

Jinlong Wang, Yunting Wu, Xiaoyun Xiong, Yuanyuan Zhang, Zhihan Lyu, Ahmed Ghoneim, and Haoran Zhao. Fedlma: A federated learning framework integrating llm-based multi-agent reasoning with knowledge distillation. *IEEE Trans. Consumer Electron.*, 71(4):11339–11349, 2025a.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

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. *Neural Information Processing Systems (NeurIPS)*, 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 (ACL)*, pp. 24013–24035, 2025b.

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. *Neural Information Processing Systems (NeurIPS)*, 35:24824–24837, 2022.

Panlong Wu, Kangshuo Li, Junbao Nan, and Fangxin Wang. Federated in-context llm agent learning. *arXiv preprint arXiv:2412.08054*, 2024.

Zhengyu Wu, Yinlin Zhu, Xunkai Li, Ziang Qiu, Rong-Hua Li, Guoren Wang, and Chenghu Zhou. Fedbook: A unified federated graph foundation codebook with intra-domain and inter-domain knowledge modeling. *arXiv preprint arXiv:2510.07755*, 2025.

Yeyu Yan, Shuai Zheng, Zhenfeng Zhu, Dong Chen, Wentao Zhang, Yawei Zhao, Kunlun He, and Yao zhao. Aligning condensed graph via hashing: A new insight for federated graph learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–15, 2026.

Rui Ye, Wenhao Wang, Jingyi Chai, Dihan Li, Zexi Li, Yinda Xu, Yaxin Du, Yanfeng Wang, and Siheng Chen. Openfedllm: Training large language models on decentralized private data via federated learning. In *the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6137–6147, 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 (ICLR)*, 2025a.

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. In *International Conference on Machine Learning (ICML)*, volume 267, 2025b.

Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. Towards building the federatedgpt: Federated instruction tuning. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6915–6919, 2024.

Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. Dense: Data-free one-shot federated learning. *Neural Information Processing Systems (NeurIPS)*, 35:21414–21428, 2022.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics (NAACL)*, pp. 2299–2314, 2024.

Jiongli Zhu, Parjanya Prashant, Alex Cloninger, and Babak Salimi. KAIROS: scalable model-agnostic data valuation. In *Neural Information Processing Systems (NeurIPS)*, 2025a.

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

Yinlin Zhu, Xunkai Li, Jishuo Jia, Miao Hu, Di Wu, and Meikang Qiu. Towards effective federated graph foundation model via mitigating knowledge entanglement. In *Neural Information Processing Systems (NeurIPS)*, 2025b.

Weiming Zhuang and Lingjuan Lyu. Fedwon: Triumphing multi-domain federated learning without normalization. In *International Conference on Learning Representations (ICLR)*, 2024.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. Gptswarm: Language agents as optimizable graphs. In *International Conference on Machine Learning (ICML)*, volume 235, pp. 62743–62767, 2024.

A Appendix

This is the appendix of our work: **"Federated Multi-Agent Orchestration via Consensus-guided Interaction Dynamics"**. The appendix is organized as follows:

A.1 Related Work.

A.2 Detailed Proofs.

A.3 Experimental Setup.

A.4 Details of FedGLV.

A.5 Additional Experiments.

A.6 Limitations and Future Work.

A.1 Related Work

A.1.1 Graph-based Multi-Agent System

Graph-based multi-agent systems model agents as nodes and their interactions as edges, providing a flexible framework for organizing collaborative reasoning. Early studies mainly rely on predefined topologies, such as chain, tree, fully connected, and debate topologies (Du et al., 2024). Although these structures are simple and reusable, they remain fixed across different queries and thus cannot adapt agent interactions to varying reasoning requirements. Motivated by this limitation, recent studies explore learnable collaboration topologies. GPTSwarm (Zhuge et al., 2024) parameterizes agent interaction graphs, while G-Designer (Zhang et al., 2025b) learns query-dependent topologies through topology generation. AgentPrune (Zhang et al., 2025a) and AgentDropout (Wang et al., 2025b) further improve collaboration efficiency by removing redundant interactions or agents from existing structures. More recently, ARG-Designer (Li et al., 2026a) generates task-specific agent compositions and communication topologies from scratch, while OFA-MAS (Li et al., 2026b) extends topology learning from domain-specific "one-for-one" models toward a unified "one-for-all" model through centralized joint training across multiple domains.

Despite these advances, existing methods are predominantly developed under centralized settings. This centralized assumption overlooks privacy constraints and the fragmented distribution of user data in the real world.

A.1.2 Federated Learning for LLM-based Agents

Federated learning (FL) has recently been extended from conventional LLM adaptation (Zhang et al., 2024; Ye et al., 2024) to agentic learning with decentralized data. FICAL (Wu et al., 2024) introduces federated in-context agent learning by exchanging compact knowledge compendiums instead of model parameters, while Fed-SE (Chen et al., 2025) enables LLM agents to perform local self-evolution across heterogeneous environments and aggregates their updates in a shared low-rank space. FedAgent (Chen et al., 2026) further investigates federated reinforcement learning for LLM agents and systematically studies robustness under task- and environment-level client heterogeneity. Moreover,

federated learning has begun to intersect with LLM-based multi-agent systems. EPEAgent (Shi et al., 2025) introduces a federated MAS paradigm and focuses on privacy-preserving information sharing during retrieval and multi-agent interaction. FedLMA (Wang et al., 2025a) incorporates LLM-based multi-agent reasoning into federated knowledge distillation to alleviate client heterogeneity, while DAWN (Wan et al., 2026) studies federated task-adaptive workflow synthesis under intra-domain difficulty skew.

However, existing federated MAS methods are largely confined to single-domain settings, i.e., clients are constructed from the same dataset and use the same agent configuration. Consequently, cross-domain learning of a universal collaboration model that accommodates heterogeneous client configurations and heterogeneous domains remains largely unexplored. Table 4 compares FedGLV with representative baselines and related methods in terms of topology design and collaborative learning.

A.1.3 Cross-Domain Knowledge Integration via Federated Learning

Beyond conventional non-IID federated learning, recent studies have explored how to integrate distributed knowledge across heterogeneous domains into shared and transferable models. FedWon (Zhuang & Lyu, 2024) studies multi-domain federated learning under domain-level distribution shifts, while FedGaLA (Pourpanah et al., 2025) further investigates federated domain generalization toward unseen domains. More recently, federated graph foundation models, such as FedGFM+ (Zhu et al., 2025b), FedBook (Wu et al., 2025), and FedRAHi (Zhu et al., 2026), establish a unified paradigm for integrating knowledge from distributed and heterogeneous graph domains into generalizable foundation models, with further efforts devoted to modeling inter-domain knowledge and improving the reliability of cross-client collaboration. These studies demonstrate the potential of federated learning to consolidate complementary knowledge across heterogeneous domains.

Inspired by these insights, we extend cross-domain federated knowledge integration to multi-agent collaboration, aiming to learn a universal collaboration model that accommodates heterogeneity arising from different client domains and agent compositions.

Table 4: Comparison of MAS orchestration methods. Query conditioning and step dependence describe topology generation. Step dependence indicates whether communication topologies vary across communication steps within a task. Multi-domain denotes joint training across task domains. Heterogeneous configurations refer to client-specific agent counts and communication steps. ✘ indicates that a property is not reported or not applicable.

	Topology Design		**Collaborative Learning**		
Method	Query Conditioning	Step Dependence	Federated Learning	Multi-Domain Joint Training	Heterogeneous Client Configs.
AgentPrune	✘	✘	✘	✘	✘
AgentDropout	✘	✔	✘	✘	✘
G-Designer	✔	✘	✘	✘	✘
ARG-Designer	✔	✘	✘	✘	✘
OFA-MAS	✔	✘	✘	✔	✘
DAWN	✔	✘	✔	✘	✘
FedGLV	✔	✔	✔	✔	✔

A.2 Detailed Proofs

Proposition 1 (Relational-State Agreement under LOO Weighting)**.** *For a fixed probe-step pair, suppress the superscripts* (m, s)*. If* $C \geq 3$*, the LOO softmax weights satisfy* $\frac{\sum_{c<l} \omega_c \omega_l \widetilde{\mathbf{x}}_c^\top \widetilde{\mathbf{x}}_l}{\sum_{c<l} \omega_c \omega_l} \geq \mathrm{U}(\mathcal{C})$*. The inequality is strict unless all client LOO values are equal.*

Proof. Fix a probe-step pair and omit its superscripts. Write $\mathrm{U} = \mathrm{U}(\mathcal{C})$ and define

$$G_{cl} = \widetilde{\mathbf{x}}_c^\top \widetilde{\mathbf{x}}_l, \qquad r_c = \omega_c - \frac{1}{C}.$$

The Gram matrix $\mathbf{G} = [G_{cl}]$ is positive semidefinite, with $G_{cc} = 1$, and $\sum_c r_c = 0$. The nonnegative relational state vectors also imply $\mathrm{U} \geq 0$. Expanding the LOO definition gives

$$v_c = \frac{2}{C-2}\left[\frac{1}{C-1}\sum_{l\neq c} G_{cl} - \mathrm{U}\right]. \tag{16}$$

Since $\omega_c = e^{v_c}/\sum_l e^{v_l}$,

$$\begin{aligned}\sum_c r_c v_c &= \sum_c r_c \log(C\omega_c) \\ &\geq 2\sum_c \frac{r_c^2}{\omega_c + 1/C} \\ &\geq \left(\sum_c |r_c|\right)^2 \geq 2\sum_c r_c^2. \end{aligned} \tag{17}$$

Here, the first inequality uses $(a-b)\log(a/b) \geq 2(a-b)^2/(a+b)$ for $a, b > 0$. The second follows from Cauchy–Schwarz, and the last uses $\sum_c r_c = 0$.

Substituting $\omega_c = 1/C + r_c$ and Eq. (16) yields

$$\begin{aligned}&\sum_{c\neq l} \omega_c\omega_l(G_{cl} - \mathrm{U}) \\ &= \frac{(C-1)(C-2)}{C}\sum_c r_c v_c + \sum_{c,l} r_c r_l G_{cl} - (1-\mathrm{U})\sum_c r_c^2 \\ &\geq \left[\frac{2(C-1)(C-2)}{C} - 1 + \mathrm{U}\right]\sum_c r_c^2 \\ &\geq \frac{1}{3}\sum_c r_c^2 \geq 0,\end{aligned}$$

where we used Eq. (17), $\mathbf{G} \succeq 0$, $C \geq 3$, and $\mathrm{U} \geq 0$.

Dividing by the positive total weight $\sum_{c\neq l} \omega_c\omega_l$ proves the above proposition. The inequality is strict whenever some $r_c \neq 0$, equivalently, whenever the LOO values are not all equal. □

A.3 Experimental Setup

A.3.1 Data Statistics

Table 5: Dataset statistics. **MCQ** denotes multiple-choice questions.

Setting	Domain	Dataset	Format	# Choices	# Train	# Test
Source Training	General Reasoning	MMLU	MCQ	4	20	219
		MMLU-Pro	MCQ	10	20	219
	Mathematical Reasoning	GSM-Hard	Numerical	-	20	219
		AQuA	MCQ	5	20	219
	Code Generation	HumanEval	Programming	-	20	141
	Graduate-level Science	GPQA	MCQ	4	20	219
Unseen Generalization	Mathematical Reasoning	SAT-Math	MCQ	4	-	119
	Medical Reasoning	MedQA	MCQ	4	-	119
	Legal Reasoning	LEXam	MCQ	4	-	119

In this work, six datasets across four domains are used for federated training: general reasoning (MMLU (Hendrycks et al., 2020) and MMLU-Pro (Wang et al., 2024)), mathematical reasoning

(GSM-Hard (Gao et al., 2023) and AQuA (Ling et al., 2017)), code generation (HumanEval (Chen et al., 2021)), and graduate-level science (GPQA (Rein et al., 2023)). For each source dataset, disjoint subsets of 20 and 219 samples are randomly selected for training and in-domain evaluation, respectively, except HumanEval, for which 141 evaluation samples are selected. To further assess cross-dataset and cross-domain generalization, we introduce three held-out datasets excluded from federated training: SAT-Math (Zhong et al., 2024) for mathematical reasoning, MedQA (Jin et al., 2021) for medical reasoning, and LEXam (Fan et al., 2026) for legal reasoning. For each held-out dataset, 119 samples are randomly selected for evaluation, with the trained model directly deployed without parameter updates or target-domain adaptation. Dataset statistics and experimental sample counts are summarized in Table 5.

A.3.2 Baseline Details

Vanilla directly prompts a single LLM to produce the final answer without explicit reasoning.

Single-Agent Reasoning Methods. We consider two representative single-agent reasoning baselines. (1) **Chain-of-Thought (CoT)** (Wei et al., 2022) prompts a single LLM to generate intermediate reasoning steps before producing the final answer. (2) **Self-Consistency with CoT (SC (CoT))** (Wang et al., 2023) generates multiple independent CoT reasoning paths and determines the final prediction through majority voting.

Fixed Multi-Agent Topologies. We evaluate several predefined communication topologies that use fixed communication patterns across different input questions. (1) **Chain** organizes agents sequentially, in which each agent receives the output of its predecessor and passes its response to the next agent. (2) **Tree** organizes agents in a hierarchical structure, allowing intermediate reasoning results to be progressively aggregated along the tree. (3) **LLM-Debate** (Du et al., 2024) performs multi-round discussion in which agents iteratively examine and respond to the reasoning of other agents before producing the final decision.

Adaptive Graph-Based Collaboration Methods. We further compare FedGLV with five recent methods that automatically optimize multi-agent communication topologies. (1) **AgentPrune** (Zhang et al., 2025a) introduces trainable masks and removes redundant message-passing edges, thereby constructing sparse communication topologies. (2) **AgentDropout** (Wang et al., 2025b) dynamically adjusts the participating agents across multi-agent communication steps through node dropout and prunes low-contribution interactions through edge dropout. (3) **G-Designer** (Zhang et al., 2025b) models agents together with a task-specific virtual node and employs a variational graph auto-encoder to generate task-adaptive communication topologies conditioned on the input query. (4) **ARG-Designer** (Li et al., 2026a) formulates multi-agent topology design as conditional autoregressive graph generation. Given an input query, it sequentially determines the required agents and generates communication links to construct a task-specific collaboration graph from scratch. (5) **OFA-MAS** (Li et al., 2026b) adopts a one-for-all topology design paradigm and learns a unified graph generator across multiple task domains using a shared pool of 19 candidate agent roles.

Federated Extensions. To support the federated setting, we further construct federated variants of G-Designer and ARG-Designer, denoted as **GDesigner**$_F$ and **ARGDesigner**$_F$, respectively. Each source client trains its topology designer on local data, and the server aggregates the shared controller parameters using sample-size-weighted FedAvg (McMahan et al., 2017) after each federated round. To ensure parameter compatibility, we use a common controller architecture and consistent parameter dimensions across clients within each baseline. Their execution settings are standardized to five worker agents and three communication steps per client. For GDesigner$_F$, we aggregate the GCN and MLP modules while retaining client-specific agent configurations and graph templates. For ARGDesigner$_F$, we align role identifiers through a shared role vocabulary and use compatible graph-encoding dimensions. Client-specific role masks restrict generation to locally available roles. In contrast to these uniform execution settings, FedGLV accommodates heterogeneous configurations across clients within a common parameterization.

A.3.3 Implementation Details

Common Experimental Settings. For a fair comparison, all experiments use GPT-4o-mini as the underlying language model, with the generation temperature fixed at 0.2 for all methods. We use the same dataset versions, sampled questions, data splits, answer normalization rules, and

evaluation scripts for comparable methods. For semantic encoding $\mathcal{T}(\cdot)$, we employ the frozen `all-MiniLM-L6-v2` Sentence-Transformer encoder, which produces 384-dimensional embeddings for task queries and agent role descriptions.

Federated Setting. We formulate the six source datasets as six heterogeneous clients: GSM-Hard, MMLU, MMLU-Pro, GPQA, AQuA, and HumanEval. Each client contains 20 local training samples. For each federated round, the server broadcasts the current global orchestrator parameters to all six clients. Each client performs local updates using only its private training samples and returns the updated parameters to the server. For GDesigner$_F$ and ARGDesigner$_F$, the server applies sample-count-weighted FedAvg. For FedGLV, parameter averaging initializes the global orchestrator, which is subsequently optimized through structural consensus distillation. No task text, agent response, reward, or generated topology is transmitted to the federated server. For GDesigner$_F$ and ARGDesigner$_F$, each client preserves its original local agent composition and domain-specific prompts. Only orchestrator parameters are communicated. The local policy-gradient reward is computed from the final answer correctness and remains inside the client.

Evaluation on Unseen Clients. For federated methods, the final global orchestrator is directly deployed to all three unseen clients. Similarly, OFA-MAS uses its single multi-domain orchestrator. For domain-specific baselines, source models are selected through a fixed correspondence based on domain relevance and reasoning characteristics: the AQuA-trained model is evaluated on SAT-Math, the MMLU-trained model on MedQA, and the MMLU-Pro-trained model on LEXam.

FedGLV Settings. The local orchestrator uses a hidden dimension of 256 and consists of two fully connected layers with GELU activations and LayerNorm, followed by a state head f_x, a growth-rate head f_b, and an edge head f_e. The relational-state and growth-rate heads predict the positive initial agent relational states and growth rates, respectively, while a state-conditioned edge head predicts a Bernoulli communication probability for each directed non-self agent pair. Communication edges are sampled during local training and selected using a default probability threshold of 0.55 during inference. Clients use $S_c = 3$ for GSM-Hard, MMLU, GPQA, and AQuA, and $S_c = 2$ for MMLU-Pro and HumanEval. All server probes use $S^m = 3$. Unless otherwise specified, inference on SAT-Math, MedQA, and LEXam uses two communication steps. Following the final-decision mechanism of G-Designer (Zhang et al., 2025b), each client includes a dedicated decision agent in addition to the five worker agents unless otherwise specified. After the prescribed communication steps, this agent receives the original task and the final responses of all worker agents and produces the client's final answer. The default sparsity regularization coefficient is $\lambda_s = 0.05$. FedGLV is trained for five federated rounds using Adam with a local learning rate of 10^{-4} and a batch size of 5. For structural aggregation, the server employs an answer-free probe bank containing 40 shared probes. The server performs five optimization epochs in each federated round.

Experimental Environment. The experiments are conducted on a machine equipped with Intel(R) Xeon(R) Gold 6248R CPU @ 3.00 GHz, and NVIDIA GeForce RTX 3090 with 24 GB of memory and CUDA 11.8. The system runs Ubuntu 20.04.2 and has 512 GB of RAM.

A.4 Details of FedGLV

A.4.1 Agent Roles

Each client is assigned five domain-specific agent roles by default, as summarized in Table 6. To encourage complementary reasoning, each role is instantiated through a specialized prompt specifying its reasoning focus and verification protocol.

Heterogeneous Agent Configurations. To evaluate FedGLV under varying agent populations (Figs. 3 and 6), we consider nested configurations with 3, 5, and 7 worker agents on GSM-Hard, MMLU-Pro, and MedQA. The 3-agent configuration retains three roles from the standard five-role configuration in Table 6, while the 7-agent configuration extends it with two complementary roles. Table 7 details these configurations. Agent counts exclude the final decision agent.

Table 6: Agent role configurations for different clients.

Dataset	Agent Roles
Source Clients	
GSM-Hard	Stepwise Calculator; Quantity Relation Parser; Equation Executor; Unit Consistency Auditor; Arithmetic Verifier.
MMLU	Evidence Mapper; Concept Boundary Analyst; Option Eliminator; Fact Checker; Reasoning Trap Auditor.
MMLU-Pro	Evidence Mapper; Constraint Solver; Controlling Rule Analyst; Reasoning Chain Auditor; Near-Miss Option Critic.
GPQA	Chemistry Transformation Analyst; Physics Model Solver; Molecular Biology Mechanism Analyst; Quantitative and Unit Auditor; Option Falsification Critic.
AQuA	Quantitative Modeler; Stepwise Solver; Arithmetic Verifier; Option Mapping Auditor; Distractor Critic.
HumanEval	Specification Parser; Algorithm Planner; Function Implementer; Edge Case Tester; Code Repairer.
Unseen Clients	
SAT-Math	Algebraic Structure Analyst; Geometry Reconstruction Solver; Data and Quantitative Reasoner; Choice-Driven Strategy Solver; Distractor and Error Auditor.
MedQA	Clinical Vignette Extractor; Pathophysiology Mechanism Analyst; Differential Diagnosis Solver; Management and Guideline Analyst; Option Safety Auditor.
LEXam	Swiss Private and Civil Law Specialist; Criminal Law and Procedure Specialist; Public Constitutional and International Law Specialist; Tax Corporate and Commercial Law Specialist; Multilingual Legal Logic and Option Auditor.

Table 7: Nested agent role configurations. The 5-agent configuration combines the first two role columns, and the 7-agent configuration uses all three.

Dataset	3-agent core	Added for 5 agents	Added for 7 agents
GSM-Hard	Quantity Relation Parser Equation Executor Arithmetic Verifier	Stepwise Calculator Unit Consistency Auditor	Reverse Consistency Solver Alternative Representation Solver
MMLU-Pro	Evidence Mapper Constraint Solver Near-Miss Option Critic	Controlling Rule Analyst Reasoning Chain Auditor	Counterexample Analyst Alternative Derivation Solver
MedQA	Clinical Vignette Extractor Differential Diagnosis Solver Management and Guideline Analyst	Pathophysiology Mechanism Analyst Option Safety Auditor	Diagnostic Test Interpreter Temporal Causal Auditor

A.4.2 Prompt Templates

Agent Role Prompts. To encourage complementary reasoning behaviors, each agent is assigned a specialized role prompt that specifies its reasoning focus and verification protocol. Prompts for two representative roles are provided below.

Option Falsification Critic

```
Falsify each supplied option against the exact stem, evidence, assumptions, and
scientific boundary conditions.

Specialist protocol:
Compare all supplied choices, preserve qualifiers, expose hidden assumptions, and
identify the exact failure of every near-miss option.
```

Option Eliminator

```
An independent solver specializing in exhaustive option-by-option comparison and
near-miss distractor rejection.

Specialist verification:
Evaluate each option separately.  Reject each losing option using a specific
contradiction, scope error, missing condition, or numerical mismatch.
```

Prompt Template for Agent Communication. FedGLV uses binary directed edges to determine which peer responses are available to each agent. All active edges share the same communication prompt. Incoming responses are presented as optional references, and the receiving agent is instructed to verify them independently before revising its answer. The following instruction is prepended once to the selected peer responses.

Communication Prompt

```
The following responses are provided by other agents as optional references.
Independently verify their reasoning and use any useful information to improve your
answer. Do not assume that these responses are correct. Preserve your answer unless
the evidence justifies a revision.

Agent {agent_id}:
{response}

...
```

Here, {agent_id} identifies a source agent, and {response} denotes its latest available response. The agent–response block is repeated for each active incoming edge.

Stepwise Prompt Construction. All agents first answer the task independently. At each subsequent communication step, the current communication topology determines which peer responses are available for each agent. For agents with at least one incoming edge, these responses are appended to the task prompt, while the role-specific instructions and task-specific prompting protocol are retained. Agents without incoming edges retain their existing responses. Responses are updated synchronously using the peer outputs available at the beginning of the step. The following schematic example illustrates the task and communication context.

Illustrative Communication-Step Prompt

Role focus:
Independently recompute the result and check the arithmetic.

Current task:
A store sells 3 notebooks for $12. How much does 1 notebook cost?

The following responses are provided by other agents as optional references.
Independently verify their reasoning and use any useful information to improve your
answer. Do not assume that these responses are correct. Preserve your answer unless
the evidence justifies a revision.

Agent 1:
The problem asks for the unit price. Divide the total cost by the number of notebooks:
$12/3 = 4$. The answer is 4.

Agent 2:
Let x be the price of one notebook. Then $3x = 12$, so $x = 4$. The answer is 4.

A.4.3 Probe Bank Construction

The probe bank is constructed independently on the server using generic task-domain and reasoning-type templates, without accessing or deriving its contents from any client-specific task instances, reference answers, or agent conversations. It contains 40 answer-free probes organized into five broad domains: mathematical reasoning, knowledge-based and multi-hop reasoning, scientific reasoning, programming, and document comprehension. Each probe pairs an abstract task description with five complementary agent role descriptions. No reference answers or correctness labels are included.

Role Configuration. Each probe is paired with five complementary role descriptions tailored to its reasoning category. For example, mathematical probes specify roles for problem formulation, solution derivation, numerical verification, distractor analysis, and consistency checking. Probes within the same category share a role configuration. For any given probe, all orchestrators receive the same ordered role descriptions, ensuring that corresponding graph nodes have matching semantic inputs.

A.4.4 Algorithm

Algorithm 1 summarizes the training procedure of FedGLV. At each federated round, clients initialize their local orchestrators from the previous global orchestrator and optimize GLV-inspired topology generation using private task feedback. The server then evaluates the updated orchestrators on the shared probe bank, constructs an LOO-weighted structural consensus, and distills it into the global orchestrator for the next round.

Algorithm 1 FedGLV: Federated Learning of a Unified MAS Orchestrator

Input: client-local datasets and configurations $\{(\mathcal{D}_c, \mathcal{R}_c, S_c)\}_{c=1}^{C}$, $C \geq 3$; frozen encoder $\mathcal{T}$; probe bank $\mathcal{B}$; initial model $[\Theta_g]^0$; federated rounds T.

for $\tau = 1, \ldots, T$ **do**
  / Client: Semantic Amortization for Dynamic MAS Orchestration */*
  for all clients $c = 1, \ldots, C$ **in parallel do**
    Receive and initialize $[\Theta_c]^\tau \leftarrow [\Theta_g]^{\tau-1}$
    for each local optimization step **do**
      For local tasks $(q, y) \sim \mathcal{D}_c$, initialize $\mathbf{x}_c^{q,0}$ and $\{b_{c,i}^q\}_{i=1}^{N_c}$ via Eq. (5)
      for $s = 0, \ldots, S_c - 1$ **do**
        Generate $\mathbf{P}_c^{q,s}$ via Eq. (7) and sample Bernoulli edges $\mathbf{A}_c^{q,s}$
        Evolve $\mathbf{x}_c^{q,s+1}$ under $\mathbf{A}_c^{q,s}$ via Eq. (8)
      end for
      Execute MAS with $\{\mathbf{A}_c^{q,s}\}_{s=0}^{S_c-1}$ and obtain task rewards R_c^q
      Update $[\Theta_c]^\tau$ via $\mathcal{L}_c$ in Eq. (11)
    end for
    Upload $[\Theta_c]^\tau$ to the server
  end for
  / Server: Data-free Structural Consensus Distillation */*
  for all probes $B^m \in \mathcal{B}$ **do**
    Evaluate $\{[\Theta_c]^\tau\}_{c=1}^{C}$ on probe B^m to obtain $\{\mathbf{x}_c^{m,s}, \mathbf{P}_c^{m,s}\}_{c,s}$
    for $s = 0, \ldots, S^m - 1$ **do**
      Compute $\{v_c^{m,s}\}_{c=1}^{C}$ via Eq. (13)
      Generate $\tilde{\mathbf{P}}^{m,s}$ according to Eq. (14)
    end for
  end for
  Obtain $[\Theta_g]^\tau$ by minimizing Eq. (15)
end for
return Global orchestrator $[\Theta_g]^T$

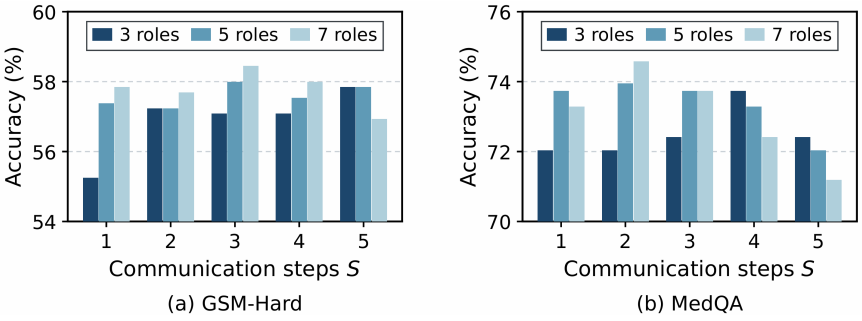


Figure 6: Performance of FedGLV under heterogeneous configurations.

A.5 Additional Experiments

A.5.1 Additional Results for Heterogeneous Agent Configurations

To complement the MMLU-Pro analysis in the main text, Fig. 6 presents additional results on GSM-Hard and MedQA, using $\{3, 5, 7\}$ worker agents and communication steps $S \in \{1, \ldots, 5\}$. The seven-role configuration achieves the highest mean accuracy on both datasets, reaching 58.45% at $S = 3$ on GSM-Hard and 74.58% at $S = 2$ on MedQA. Further extending communication to $S = 5$ reduces these accuracies to 56.93% and 71.19%, respectively, consistent with the observation that prolonged communication does not necessarily improve performance. The benefit of additional roles also depends on the communication setting. For example, on MedQA at $S = 4$, three roles achieve 73.73% accuracy, compared with 72.41% for seven roles. Thus, the preferred configuration varies across datasets, and increasing the agent population alone does not guarantee better performance. These additional results on a source client and an unseen client further support FedGLV's applicability across heterogeneous configurations and reinforce the importance of jointly considering role diversity and communication steps.

(MedQA) Query:
A 24-year-old woman presents with urinary retention, a history of transient speech and visual disturbances, recent facial pain, and abnormal conjugate gaze. What is the best initial test?
Options:
(A) Head CT (B) Head MRI (C) Lumbar puncture (D) Domestic abuse screening and assessment of life stressors

Initial

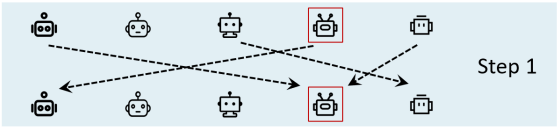
Agent 1: The combination of neurological symptoms suggests a central nervous system disorder. MRI is better suited to evaluating possible demyelination. **(B)**

Agent 2: The patient's young age and episodic neurological symptoms raise suspicion for multiple sclerosis. MRI can identify associated demyelinating lesions more sensitively than CT. **(B)**

Agent 3: CT is useful for hemorrhage or mass effect, whereas MRI is more sensitive to demyelination and subtle lesions relevant to this presentation. **(B)**

Agent 4: Although demyelination is possible, emergency evaluation should first exclude acute intracranial pathology, such as hemorrhage or mass effect. Head CT is therefore the preferred initial test. **(A)**

Agent 5: MRI provides a more suitable evaluation of the suspected neurological pathology than the other available options. **(B)**



After Communication

Agent 1: The neurological findings warrant evaluation for central nervous system pathology. MRI is more sensitive than CT for detecting possible demyelinating lesions. **(B)**

Agent 4: MRI is the best initial test for detecting demyelinating lesions that could explain this presentation. Although CT can help exclude hemorrhage, it is less sensitive to the lesions relevant to this case. **(B)**

Agent 5: Transient speech and visual disturbances, together with facial pain, suggest a possible demyelinating condition. MRI is the most appropriate initial test to investigate these potential lesions. **(B)**

Figure 7: Selective communication and local error correction by FedGLV on MedQA.

(MedQA) Query:
A 36-year-old man develops a postoperative wound infection. Culture reveals gram-positive cocci in clusters. Fever and purulent discharge persist despite dicloxacillin. Which bacterial characteristic best explains treatment failure?
Options:
(A) Presence of an impenetrable outer membrane (B) Ability to cleave β-lactam rings
(C) Development of efflux pumps **(D) Adaptation in binding proteins**

Initial

Agent 1: The organism is likely \textit{Staphylococcus aureus}. Production of β-lactamase could inactivate the antibiotic and explain treatment failure. **(B)**

Agent 2: Although changes in binding proteins can confer resistance, cleavage of β-lactam rings is the most plausible explanation for the failure of dicloxacillin. **(B)**

Agent 3: The lack of improvement with dicloxacillin suggests MRSA. Changes in penicillin-binding proteins reduce antibioticbinding and explain resistance. **(D)**

Agent 4: Dicloxacillin is effective against susceptible \textit{S. aureus}. Persistent infection suggests resistanceassociated with altered penicillin-binding proteins. **(D)**

Agent 5: Both β-lactamase production and altered binding proteins are plausible, but cleavage of β-lactamrings is the more direct explanation. **(B)**

After 2 Communication Steps

Agent 1: The failure of dicloxacillin suggests MRSA.Altered binding proteins provide the most likely explanation. **(D)**

Agent 2: Altered binding proteins are relevant to MRSA, but β-lactamase production more directly explains the ineffectiveness of the antibiotic. **(B)**

Agent 3: MRSA-associated changes in binding proteins best explainthe persistent infection despite dicloxacillin treatment. **(D)**

Agent 4: Altered binding proteins reduce the affinity of β-lactam antibiotics and explain treatment failurein the suspected MRSA infection. **(D)**

Agent 5: Given the suspected MRSA infection, adaptation in bindingproteins is the most relevant mechanism explaining dicloxacillin failure. **(D)**

Figure 8: Correction of an initially incorrect majority after communication on MedQA.

A.5.2 Case Study

Selective Information Routing. Fig. 7 illustrates communication through a sparse topology with four directed links among five agents. Agent 4 initially selects Head CT (A), while the other agents favor Head MRI (B). After receiving references from Agents 1 and 5, Agent 4 revises its answer to B. Meanwhile, Agent 1 retains its correct answer despite receiving Agent 4's initially incorrect response.

This case illustrates how selective communication can provide useful references for local correction while preserving independent judgment.

Correction of an Incorrect Majority. Fig. 8 examines whether collaboration can revise an initially incorrect majority. Before communication, Agents 1, 2, and 5 select B, whereas only Agents 3 and 4 identify the correct answer D. After communication, Agents 1 and 5 revise their answers to D, citing altered binding proteins as the explanation for treatment failure. Consequently, support for the correct answer increases from two agents to four, although Agent 2 retains its incorrect answer. This example illustrates how exchanging reasoning can overturn an incorrect initial majority without requiring unanimous agreement.

A.6 Limitations and Future Work

Despite these strengths, FedGLV has several limitations that leave room for improvement. For example, classical GLV models consider mutualistic and competitive relationships between species when modeling ecological systems, with this distinction reflected in the signs of the interaction coefficients. Our work focuses on nonnegative interactions among agents, a simplification also commonly adopted by existing MAS methods. Future work could extend FedGLV to explicitly incorporate signed edges as signals of evidence support and rebuttal, thereby enhancing collaborative reasoning among agents. Moreover, LOO-guided valuation provides an answer-free measure of client contributions to relational-state agreement, although it may not fully capture task-specific relevance without access to additional client-specific information. Therefore, selectively sharing auxiliary information from each client could offer finer guidance for structural consensus. Nonetheless, as the core contribution of this work is to establish a unified federated framework for transferable multi-agent orchestration, we leave these extensions to future research.