Heterogeneity-Aware Knowledge Sharing for Graph Federated Learning

Wentao Yu [1] **Sheng Wan** [2] **Shuo Chen** [3 4] **Bo Han** [5 4] **Chen Gong** [6]

Abstract

Graph Federated Learning (GFL) enables distributed graph representation learning while preserving graph data privacy. However, it suffers from heterogeneity in node features and graph structures across clients. To address this challenge, we propose a novel graph **Fed**erated learning method via **S**emantic and **S**tructural **A**lignment (FedSSA). For node feature heterogeneity, FedSSA infers class-wise node distributions through a variational model, clusters clients according to the inferred distributions, and aligns local distributions with cluster-level representatives. For structural heterogeneity, FedSSA employs spectral Graph Neural Networks (GNNs) and introduces a spectral energy measure to characterize graph topology, enabling structural alignment between local and cluster-level spectral GNNs. Experiments on eleven homophilic and heterophilic graph datasets under non-overlapping and overlapping partitioning settings demonstrate that FedSSA consistently outperforms eleven state-of-the-art methods. Our code is available at https://github.com/blgpb/FedSSA.

1. Introduction

Graphs are fundamental data structures in real-world applications, such as social networks, transportation systems, and molecular chemistry (Bai et al., 2022; Yu et al., 2023; Zhou et al., 2024; 2025; 2026; Yu et al., 2026a). In many real-world applications, large-scale graphs are often partitioned into a set of subgraphs and distributed across multiple clients due to storage and computation limitations (Meng et al., 2024). In addition, subgraphs are only locally accessible due to restrictions from privacy regulations and data protection protocols. Therefore, Graph Federated Learning (GFL) has emerged as a promising paradigm for distributed graph representation learning, where multiple clients collaboratively train models without sharing their raw graph data (Baek et al., 2023; Yu, 2025; Yu et al., 2025b; 2026b).

However, due to different distributions of graph data across clients, graph data on each client is usually non-independent and identically distributed (non-IID). Consequently, GFL suffers from heterogeneity arising from both diverse node features and varied structural topologies across clients (Li et al., 2023; 2024; Zhu et al., 2024). These two types of heterogeneity pose significant challenges to GFL, leading to training instability and performance degradation (Huang et al., 2024b). To tackle this fundamental and challenging issue, a number of studies have been proposed. Specifically, existing methods can be broadly categorized into two types: (i) methods that solely focus on addressing structural heterogeneity, and (ii) methods that do not differentiate between node feature heterogeneity and structural heterogeneity. For the first type, representative methods include FedGTA (Li et al., 2023), FedTAD (Zhu et al., 2024), and AdaFGL (Li et al., 2024), which perform topology-aware aggregation by considering structural information from other clients. However, these methods are based on homophily assumption (*i.e.*, edges tend to connect similar nodes), which does not hold in heterophilic graphs (Platonov et al., 2023a; Yu, 2025). For the second type, typical methods include GCFL (Xie et al., 2021), FED-PUB (Baek et al., 2023), and FedIIH (Yu et al., 2025a), which primarily carry out personalized federated aggregation by adjusting aggregation weights among clients. However, simply performing the weighted aggregation of model parameters may not be truly effective in solving heterogeneity, since model parameters may fail to represent the intrinsic characteristics of node features and structural topologies, thereby restricting their effectiveness in scenarios with strong heterogeneity.

Therefore, we pose the following research question:

How can we explicitly address node feature heterogeneity and structural heterogeneity in graph federated learning?

[1]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu, China [2]College of Smart Agriculture, Nanjing Agricultural University, Nanjing, Jiangsu, China [3]School of Intelligence Science and Technology, Nanjing University, Suzhou, Jiangsu, China [4]Center for Advanced Intelligence Project, RIKEN, Chuo-ku, Tokyo, Japan [5]Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong, China [6]School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai, China. Correspondence to: Chen Gong <chen.gong@sjtu.edu.cn>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

To answer this question, we propose a novel graph **Fed**erated learning method via **S**emantic and **S**tructural **A**lignment (FedSSA), which shares the knowledge of both node features and structural topologies among clients to tackle two types of heterogeneity in GFL. On one hand, to address node feature heterogeneity, we propose a novel variational model to infer the class-wise distributions of node features on each client. By clustering clients based on these inferred distributions, we construct cluster-level representative distributions for each class by matching statistical moments (*i.e.*, mean and covariance). Afterwards, we enforce class-wise semantic knowledge alignment by minimizing the divergence between local distributions and cluster-level distributions to facilitate semantic knowledge sharing. On the other hand, to address structural heterogeneity, we employ spectral Graph Neural Networks (GNNs) and propose a novel spectral energy measure to characterize the structural information of each client. By clustering clients based on their spectral energies embedded in Grassmann manifold, we construct a cluster-level spectral GNN for each cluster. Subsequently, we align the spectral characteristics of local spectral GNNs with those of cluster-level spectral GNNs to enable structural knowledge sharing. Furthermore, we theoretically prove that our proposed FedSSA converges at a linear rate. Extensive experiments on eleven datasets demonstrate the effectiveness of our FedSSA. To be specific, it outperforms the second-best method by a large margin of 2.82% in terms of classification accuracy.

Conflict of Interest Disclosure The authors declare no financial conflicts of interest related to this work.

2. Related Work

In this section, we review the typical works related to this paper, including Graph Federated Learning (GFL), Clustered Federated Learning (CFL), and Spectral GNNs.

2.1. Graph Federated Learning

Graph Federated Learning (GFL) aims to train GNNs across multiple clients without sharing raw graph data (Baek et al., 2023; Yu, 2025; Yu et al., 2025b). However, due to the non-IID nature of graph data across clients, GFL faces significant challenges stemming from heterogeneity in both node features and structural topologies (Li et al., 2023; 2024; Zhu et al., 2024). To tackle this issue, existing methods can be broadly categorized into two types: (i) methods that solely focus on mitigating structural heterogeneity, and (ii) methods that do not differentiate between node feature heterogeneity and structural heterogeneity.

For the first type, representative methods include FedGTA (Li et al., 2023), FedTAD (Zhu et al., 2024), and AdaFGL (Li et al., 2024), which modifies local models with structural information from other clients. For example, FedGTA (Li et al., 2023) recalibrates aggregation weights based on topology-aware local smoothing confidence. Subsequently, FedTAD (Zhu et al., 2024) employs topology-aware knowledge distillation to rectify unreliable knowledge arising from structural heterogeneity. Meanwhile, AdaFGL (Li et al., 2024) leverages federated knowledge extractor to optimize structural topology on each client. However, these methods are based on homophily assumption (*i.e.*, edges tend to connect similar nodes), which does not hold in heterophilic graphs (Platonov et al., 2023a).

For the second type, typical methods include GCFL (Xie et al., 2021), FED-PUB (Baek et al., 2023), and FedIIH (Yu et al., 2025a), which treats node feature heterogeneity and structural heterogeneity as a unified challenge. These methods mainly carry out personalized federated aggregation by adjusting the aggregation weights of clients. For example, GCFL (Xie et al., 2021) dynamically clusters clients and aggregates model parameters within clusters to mitigate heterogeneity. Alternatively, FED-PUB (Baek et al., 2023) optimizes the aggregation weights based on similarities between pairwise clients, which are calculated based on the outputs of local GNNs. Similarly, FedIIH (Yu et al., 2025a) performs a weighted federation of model parameters based on similarities calculated from inferred graph distributions. Nevertheless, in practice, node feature heterogeneity and structural heterogeneity have different characteristics. Consequently, simply performing the weighted aggregation of model parameters may not be truly effective in solving heterogeneity, as model parameters may fail to reveal the intrinsic characteristics of node features and structural topologies. This limitation significantly restricts their performance in scenarios with strong heterogeneity. Unlike existing methods, we propose to separately address node feature heterogeneity and structural heterogeneity.

2.2. Clustered Federated Learning

Clustered Federated Learning (CFL) aims to mitigate heterogeneity by grouping clients with similar data distributions into clusters and constructing personalized models tailored for each cluster (Sattler et al., 2021; Ghosh et al., 2022; Kim et al., 2024; Vardhan et al., 2024). Most of the existing CFL methods mainly leverage three types of clustering signals, namely gradients (Sattler et al., 2021), parameters (Vardhan et al., 2024), and losses (Ghosh et al., 2022). Specifically, CFL (Sattler et al., 2021) adopts a top-down mechanism to recursively bipartition clients based on the cosine similarity of gradients. In contrast, SR-FCA (Vardhan et al., 2024) utilizes a bottom-up mechanism by initializing distinct clusters for each client and then successively refining them based on the similarity of parameters. Alternatively, IFCA (Ghosh et al., 2022) broadcasts K models to each client. Each client then evaluates K models and selects the model that yields

the lowest loss to determine its cluster assignment. However, these methods are primarily tailored for traditional federated learning scenarios and fail to incorporate the structural topologies of graph data. Consequently, our work proposes a novel clustering mechanism by explicitly leveraging node features and structural topologies, respectively.

2.3. Spectral Graph Neural Networks

Spectral GNNs leverage the spectral properties of the graph Laplacian to perform graph signal filtering in the spectral domain (Defferrard et al., 2016; He et al., 2021; Guo & Wei, 2023; Chen et al., 2024). Fundamentally, these methods utilize the eigenvalues and eigenvectors of the graph Laplacian matrix to define graph spectral filters that capture and model structural information. For instance, early approaches such as ChebNet (Defferrard et al., 2016) employ Chebyshev polynomial bases to efficiently approximate graph spectral filters. However, conventional spectral GNNs typically optimize polynomial coefficients without sufficient constraints, which leads to ill-posed filters. To address this problem, BernNet (He et al., 2021) constrains the coefficients of the Bernstein basis to ensure the learned filters accurately reflect the spectral characteristics of observed graph. Subsequently, OptBasisGNN (Guo & Wei, 2023) further improves the expressive power of spectral graph filters by learning an optimal polynomial basis directly from graph data. Motivated by the capability of Spectral GNNs to capture structural information, we employ spectral GNNs to characterize structural topologies and facilitate structural knowledge sharing.

3. Notations

This section introduces mathematical notations used throughout the paper. We focus on node classification task in GFL, where M clients collaboratively train GNNs on their local graphs. Let $\mathcal{M} = \{1, 2, \cdots, M\}$ denote the set of clients. Each client $m \in \mathcal{M}$ holds a local graph $\mathcal{G}_m = \langle \mathcal{V}_m, \mathcal{E}_m \rangle$, where $\mathcal{V}_m$ and $\mathcal{E}_m$ denote the node set and edge set, respectively. For client m, let $n_m = |\mathcal{V}_m|$ and $e_m = |\mathcal{E}_m|$ denote the number of nodes and the number of edges, while d and C represent the feature dimension and the number of classes, respectively. In addition, we denote node feature matrix as $\mathbf{X}_m \in \mathbb{R}^{n_m \times d}$, adjacency matrix as $\mathbf{A}_m \in \mathbb{R}^{n_m \times n_m}$, and label matrix as $\mathbf{Y}_m \in \mathbb{R}^{n_m \times 1}$. Furthermore, let $\mathbf{L}_m = \mathbf{I} - \mathbf{D}_m^{-\frac{1}{2}} \mathbf{A}_m \mathbf{D}_m^{-\frac{1}{2}}$ denote the symmetric normalized Laplacian matrix of $\mathcal{G}_m$ without self-loops, where $\mathbf{I}$ is the identity matrix and $\mathbf{D}_m \in \mathbb{R}^{n_m \times n_m}$ is the degree matrix. Subsequently, let $\mathbf{L}_m = \mathbf{U}_m \mathbf{\Lambda}_m \mathbf{U}_m^\top$ denote the eigendecomposition of $\mathbf{L}_m$, where $\mathbf{U}_m$ represents the matrix of eigenvectors and $\mathbf{\Lambda}_m$ denotes the diagonal matrix of eigenvalues. Therefore, the graph Fourier transform of $\mathbf{X}_m$ can be defined as $\tilde{\mathbf{X}}_m = \mathbf{U}_m^\top \mathbf{X}_m \in \mathbb{R}^{n_m \times d}$.

Here we also introduce some norms that will be used. First, $\|\mathbf{v}\|_2 = \sqrt{\sum_i v_i^2}$ denotes the ℓ_2-norm of a vector $\mathbf{v}$, where v_i is the i-th element of $\mathbf{v}$. Second, $\|\mathbf{X}\|_{\mathrm{F}} = \sqrt{\sum_{i,j} |x_{ij}|^2}$ denotes the Frobenius norm of a matrix $\mathbf{X}$, where x_{ij} is the (i, j)-th element of $\mathbf{X}$.

4. Our Proposed Method

In this section, we provide the details of our proposed graph **Fed**erated learning method via **S**emantic and **S**tructural **A**lignment (FedSSA).

4.1. Framework Overview

As shown in Figure 1, our proposed FedSSA comprises two components. **(i) Semantic Knowledge Sharing:** To address the heterogeneity of node feature, we propose a variational model to infer class-wise node feature distributions on each client. Clients are clustered based on inferred distributions, and local distributions are aligned with cluster-level distributions. **(ii) Structural Knowledge Sharing:** To address structural heterogeneity, we employ spectral GNNs and introduce a spectral energy measure to characterize local structural topologies. Clients are clustered by spectral energy, which enables the alignment between local spectral GNNs and cluster-level spectral GNNs.

4.2. Semantic Knowledge Sharing

To address node feature heterogeneity, we aim to share semantic knowledge among clients.

Variational Model We first propose a variational model to infer the class-wise distributions of node features on each client. As illustrated in Figure 2, $\mathbf{X}_m$ and $\mathbf{Y}_m$ are observed variables, while $\mathbf{Z}_m$ represents a latent variable on the m-th client. Based on the graphical model in Figure 2, the joint probability distribution can be factorized as follows:

$$p(\mathbf{X}_m, \mathbf{Y}_m, \mathbf{Z}_m) = p(\mathbf{X}_m | \mathbf{Z}_m, \mathbf{Y}_m) p(\mathbf{Z}_m) p(\mathbf{Y}_m), \quad (1)$$

where $p(\mathbf{Z}_m)$ denotes the prior distribution of $\mathbf{Z}_m$, and $p(\mathbf{X}_m | \mathbf{Z}_m, \mathbf{Y}_m)$ denotes the conditional distribution. We then employ the true posterior distribution $p(\mathbf{Z}_m | \mathbf{X}_m, \mathbf{Y}_m)$ to represent the class-wise distributions of node features. However, this true posterior inference is computationally intractable. Therefore, we attempt to infer it with a variational distribution $q(\mathbf{Z}_m | \mathbf{X}_m, \mathbf{Y}_m)$ via variational inference (Kingma & Welling, 2013; Kingma et al., 2014). According to the graphical model in Figure 2, the Evidence Lower BOund (ELBO) can be formulated as follows:

$$\begin{aligned} \mathcal{L}_{\mathrm{ELBO}}(\mathbf{X}_m, \mathbf{Y}_m) = \mathbb{E}_{q(\mathbf{Z}_m | \mathbf{X}_m, \mathbf{Y}_m)} \log p(\mathbf{X}_m | \mathbf{Y}_m, \mathbf{Z}_m) \\ + \log p(\mathbf{Y}_m) \\ - D_{\mathrm{KL}}\big(q(\mathbf{Z}_m | \mathbf{X}_m, \mathbf{Y}_m) \parallel p(\mathbf{Z}_m)\big), \end{aligned} \quad (2)$$

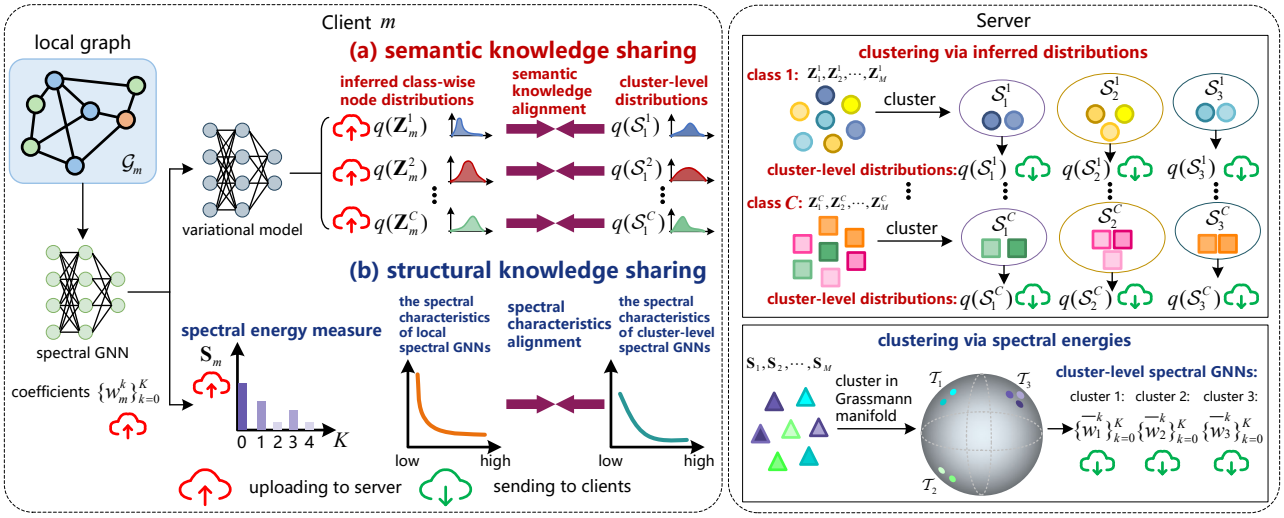


Figure 1. The overview of our proposed FedSSA. **(a) Semantic Knowledge Sharing:** A variational model infers class-wise node feature distributions on each client. Clients are clustered based on inferred distributions, and local distributions are aligned with cluster-level distributions. **(b) Structural Knowledge Sharing:** Spectral GNNs are employed alongside a spectral energy measure to characterize local structural topologies. Clients are clustered by spectral energy, which enables alignment between local and cluster-level spectral GNNs.

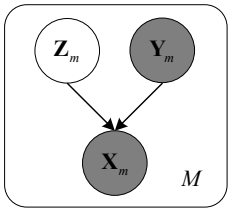


Figure 2. The graphical model of our proposed variational model, where $\mathbf{Z}_m$ is a latent variable, $\mathbf{X}_m$ and $\mathbf{Y}_m$ are observed variables.

where $\log p(\mathbf{Y}_m)$ denotes the log prior probability of $\mathbf{Y}_m$, and D_{KL} denotes Kullback-Leibler (KL) divergence. The derivation of Eq. (2) can be found in Appendix B. Consequently, the latent distribution $q(\mathbf{Z}_m^c)$ of nodes with class label c on the m-th client can be derived as follows:

$$\begin{aligned} \mathcal{I}_m^c &\triangleq \{i \in \{1, 2, \cdots, n_m\} \mid (\mathbf{Y}_m)_{[i,1]} = c\}, \\ q(\mathbf{Z}_m^c) &\triangleq q\big((\mathbf{Z}_m)_{[\mathcal{I}_m^c,:]} \,\big|\, (\mathbf{X}_m)_{[\mathcal{I}_m^c,:]}, (\mathbf{Y}_m)_{[\mathcal{I}_m^c,:]}\big), \end{aligned} \tag{3}$$

where $c = 1, 2, \cdots, C$, $(\mathbf{X}_m)_{[\mathcal{I}_m^c,:]}$, $(\mathbf{Y}_m)_{[\mathcal{I}_m^c,:]}$, and $(\mathbf{Z}_m)_{[\mathcal{I}_m^c,:]}$ denote submatrices (*i.e.*, rows) indexed by $\mathcal{I}_m^c$.

Variational Graph Autoencoder To infer latent distribution $q(\mathbf{Z}_m^c)$ in Eq. (3), we employ a Variational Graph AutoEncoder (VGAE) (Kipf & Welling, 2016) on each client. Due to space limitations, details are provided in Appendix C. Specifically, we instantiate $q(\mathbf{Z}_m^c)$ as a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_m^c, \boldsymbol{\Sigma}_m^c)$, where $\boldsymbol{\mu}_m^c$ and $\boldsymbol{\Sigma}_m^c$ denote the mean vector and covariance matrix, respectively.

Clustering via Inferred Distributions After obtaining the class-wise latent distributions of each client, we cluster clients based on these inferred distributions to facilitate semantic knowledge sharing. Specifically, for each class c, we first employ the reparameterization trick (Kingma & Welling, 2013) to obtain $\tilde{\mathbf{Z}}_m^c = \boldsymbol{\mu}_m^c + (\boldsymbol{\Sigma}_m^c)^{\frac{1}{2}} \boldsymbol{\epsilon}_m^c$, where $\boldsymbol{\epsilon}_m^c \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We then collect samples $\{\tilde{\mathbf{Z}}_m^c | m \in \mathcal{M}\}$ from all clients and cluster them into K_{node} clusters, which can be defined as

$$\{\tilde{\mathbf{Z}}_m^c\}_{m=1}^M \xrightarrow{\text{Cluster}} \{\mathcal{S}_i^c\}_{i=1}^{K_{\text{node}}}, \tag{4}$$

where $\mathcal{S}_i^c$ denotes the set of clients in the i-th cluster for class c. Therefore, we can use the Gaussian Mixture Model (GMM) to represent mixed Gaussian distributions in cluster $\mathcal{S}_i^c$ as follows:

$$p(\mathcal{S}_i^c) = \sum_{m \in \mathcal{S}_i^c} \omega_m^c \cdot \mathcal{N}(\boldsymbol{\mu}_m^c, \boldsymbol{\Sigma}_m^c), \tag{5}$$

where ω_m^c denotes the mixture weight of the m-th client in cluster $\mathcal{S}_i^c$, and $\sum_{m \in \mathcal{S}_i^c} \omega_m^c = 1$. Here, $p(\mathcal{S}_i^c)$ denotes the mixture distribution of class-c latent variables from all clients in cluster $\mathcal{S}_i^c$, *i.e.*, the mixture of $\{\mathcal{N}(\boldsymbol{\mu}_m^c, \boldsymbol{\Sigma}_m^c)\}_{m \in \mathcal{S}_i^c}$. However, the computational complexity of GMM increases with the number of clients in the cluster, which makes it inefficient. Therefore, we further simplify $p(\mathcal{S}_i^c)$ by approximating it with a single Gaussian distribution. Specifically, we construct a cluster-level representative distribution $q(\mathcal{S}_i^c) = \mathcal{N}(\boldsymbol{\mu}_i^c, \boldsymbol{\Sigma}_i^c)$ by matching the first and second moments of local distributions within the

cluster, which can be calculated as follows:

$$\begin{aligned}\boldsymbol{\mu}_i^c &= \sum_{m\in\mathcal{S}_i^c} \omega_m^c \boldsymbol{\mu}_m^c, \\ \boldsymbol{\Sigma}_i^c &= \Big(\sum_{m\in\mathcal{S}_i^c} \omega_m^c (\boldsymbol{\Sigma}_m^c + \boldsymbol{\mu}_m^c (\boldsymbol{\mu}_m^c)^\top)\Big) - \boldsymbol{\mu}_i^c (\boldsymbol{\mu}_i^c)^\top,\end{aligned} \tag{6}$$

where $\omega_m^c = \frac{n_m^c}{\sum_{n\in\mathcal{S}_i^c} n_n^c}$, n_m^c is the number of nodes with label c on client m.

Semantic Knowledge Alignment After obtaining cluster-level representative distributions, we align local distributions with cluster-level distributions to facilitate semantic knowledge sharing. Specifically, we minimize the KL divergence between local distribution $q(\mathbf{Z}_m^c)$ and cluster-level distribution $q(\mathcal{S}_i^c)$ as follows:

$$\mathcal{L}_{\text{node}} = \sum_{c=1}^{C}\sum_{i=1}^{K_{\text{node}}}\sum_{m\in\mathcal{S}_i^c} D_{\text{KL}}\big(q(\mathbf{Z}_m^c) \parallel q(\mathcal{S}_i^c)\big), \tag{7}$$

which implies that the optimization of $\mathcal{L}_{\text{node}}$ can be performed in parallel across M clients.

4.3. Structural Knowledge Sharing

To address structural heterogeneity, we aim to share structural knowledge among clients.

Spectral GNN Since spectral GNNs are constructed based on polynomial bases derived from graph Laplacian spectrum, they naturally capture structural information of graphs. Consequently, we employ a spectral GNN on each client to effectively capture structural information, which facilitates structural knowledge sharing. In this paper, we employ the polynomial filter-based spectral GNNs (*e.g.*, ChebNet (Defferrard et al., 2016), BernNet (He et al., 2021)). Typically, they can be formulated as follows:

$$\mathbf{P}_m = \sum_{k=0}^{K} w_m^k \mathbf{H}_m^k, \tag{8}$$

where $\mathbf{H}_m^k = (\mathbf{L}_m)^k \mathbf{X}_m \in \mathbb{R}^{n_m\times d}$ denotes the polynomial filters of the k-th order, $w_m^k \in \mathbb{R}$ represents the learnable coefficients of the k-th order, $\mathbf{P}_m$ denotes graph representation, and K is the number of orders. According to the theory of graph Fourier transform (Shuman et al., 2013), we can have $\mathbf{H}_m^k = \mathbf{U}_m \boldsymbol{\Lambda}_m^k \tilde{\mathbf{X}}_m$. Therefore, the physical meaning of $\mathbf{H}_m^k$ is the signal of the k-th spectral band. Meanwhile, w_m^k represents learnable coefficients of the k-th spectral band. Consequently, $\mathbf{P}_m$ can be viewed as a weighted combination of signals from different spectral bands, where weights are determined by learnable coefficients $\{w_m^k\}_{k=0}^K$. In other words, spectral GNNs effectively capture the structural information of graphs by learning to combine signals from various spectral bands (Yang et al., 2025).

Spectral Energy Measure To characterize the structural information of each client, we analyze the contribution of different spectral bands to the learned graph representation. Specifically, the term $w_m^k \mathbf{H}_m^k$ in Eq. (8) can be interpreted as the *weighted spectral response* of the k-th band. It represents graph signal explicitly activated by the learnable filter for downstream task. To derive a descriptor effectively capturing the spectral characteristics of the m-th client, we aim to propose a spectral energy measure. First, we compute the average response vector for each band via

$$\begin{aligned}\mathbf{F}_m^k &= w_m^k \mathbf{H}_m^k \in \mathbb{R}^{n_m\times d}, \\ \mathbf{E}_m^k &= \frac{1}{n_m}\sum_{i=1}^{n_m} (\mathbf{F}_m^k)_{[i,:]} \in \mathbb{R}^d,\end{aligned} \tag{9}$$

where $(\mathbf{F}_m^k)_{[i,:]}$ represents the i-th row of matrix $\mathbf{F}_m^k$. Here, $\mathbf{E}_m^k$ quantifies the *expectation of spectral response* corresponding to the k-th spectral order. A larger magnitude of $\mathbf{E}_m^k$ implies that the k-th spectral band plays a more dominant role in graph representation. Subsequently, we define the spectral energy measure $\mathbf{S}_m$ for the m-th client by concatenating these response vectors as follows:

$$\mathbf{S}_m = [\mathbf{E}_m^0, \mathbf{E}_m^1, \cdots, \mathbf{E}_m^K] \in \mathbb{R}^{d\times(K+1)}, \tag{10}$$

where $[\cdot,\cdot]$ denotes the concatenation operation. This measure $\mathbf{S}_m$ essentially acts as a *spectral fingerprint*, which reflects the energy distribution of structural topologies as learned by spectral GNN. Therefore, it can be utilized to facilitate structural knowledge sharing among clients.

Clustering via Spectral Energies After obtaining the spectral energy measure for each client, we cluster clients based on these spectral energies to facilitate structural knowledge sharing. Specifically, we collect spectral energies $\{\mathbf{S}_m | m \in \mathcal{M}\}$ from all clients. Since each spectral energy measure can be spanned as a spectral subspace, we embed them into the Grassmann manifold, which provides a natural geometric space for measuring subspace-based spectral representations. To measure the distance of spectral subspaces, we employ Chordal distance due to its effectiveness and simplicity. Specifically, we first obtain an orthonormal basis of the column space of $\mathbf{S}_m$ via QR decomposition:

$$\mathbf{S}_m = \mathbf{Q}_m \mathbf{R}_m, \quad \mathbf{Q}_m^\top \mathbf{Q}_m = \mathbf{I}_{K+1}, \tag{11}$$

where $\mathbf{I}_{K+1} \in \mathbb{R}^{(K+1)\times(K+1)}$ is an identity matrix and $\mathbf{Q}_m \in \mathbb{R}^{d\times(K+1)}$. The Chordal distance between two spectral energy measures is then calculated as

$$d_{\text{Chordal}}(\mathbf{Q}_m, \mathbf{Q}_n) = \big(K+1 - \|\mathbf{Q}_m^\top \mathbf{Q}_n\|_{\text{F}}^2\big)^{\frac{1}{2}}. \tag{12}$$

We then cluster spectral energies based on their Chordal distances into K_{struct} clusters, which can be defined as

$$\{\mathbf{S}_m\}_{m=1}^M \xrightarrow{\text{Cluster}} \{\mathcal{T}_j\}_{j=1}^{K_{\text{struct}}}, \tag{13}$$

where $\mathcal{T}_j$ denotes the set of clients in the j-th cluster.

Structural Knowledge Alignment After obtaining cluster-level spectral GNNs, we align the spectral characteristics of local spectral GNNs with those of cluster-level spectral GNNs to enable structural knowledge sharing. Specifically, for cluster $\mathcal{T}_j$, we compute the mean of learnable coefficients of spectral GNNs in $\mathcal{T}_j$ as follows:

$$\overline{w}_j^k \leftarrow \frac{1}{|\mathcal{T}_j|} \sum_{m \in \mathcal{T}_j} w_m^k. \tag{14}$$

Subsequently, we minimize the discrepancy between the coefficients of local spectral GNNs and those of cluster-level spectral GNNs as follows:

$$\mathcal{L}_{\text{align}} = \sum_{k=0}^{K} \sum_{j=1}^{K_{\text{struct}}} \sum_{m \in \mathcal{T}_j} |w_m^k - \overline{w}_j^k|. \tag{15}$$

Moreover, we add a regularization term to constrain the coefficients of spectral GNNs as:

$$\mathcal{L}_{\text{reg}} = \sum_{m=1}^{M} \sum_{k=0}^{K} \left(\lambda_1 |w_m^k| + \frac{\lambda_2}{2} (w_m^k)^2\right). \tag{16}$$

Finally, we can have $\mathcal{L}_{\text{struct}} = \mathcal{L}_{\text{align}} + \mathcal{L}_{\text{reg}}$, where $\lambda_1, \lambda_2 > 0$ are hyperparameters adjusting regularization.

4.4. Convergence Analysis

Here we analyze the convergence property of our proposed FedSSA. For client $m \in \mathcal{M}$, let $\mathbf{w}_m^t$ denote its local model parameters after t communication rounds. For simplicity, we omit subscript m if no notational confusion is incurred. In particular, $\mathbf{w}^0$ denotes the initialized local model and $\mathbf{w}^T$ denotes the local model after T communication rounds.

Assumption 4.1 (Regularity and Boundedness)**.** To facilitate theoretical analysis, we make following assumptions (Detailed descriptions are provided in Appendix D.1.):

- **Regularity of Loss Function:** The population risk function $F(\mathbf{w})$ is L_F-smooth and λ_F-strongly convex.
- **Bounded Intra-cluster Heterogeneity:** We assume the divergence within each cluster is bounded. Specifically, for semantic knowledge, the differences in variational parameters are bounded by constants δ_μ and δ_Σ, respectively. For structural knowledge, Chordal distance between spectral energy matrices is bounded by ϵ_U.
- **Properties of Learnable Components:** The variational parameters are differentiable with Jacobians bounded by L_q. Furthermore, the learnable coefficients of spectral GNNs are bounded by $w_{\max}$ and satisfy the Lipschitz continuity property with constant L_w with respect to spectral energy matrices.

Theorem 4.2 (Convergence of FedSSA)**.** *Suppose that Assumption 4.1 holds. Let $\eta = \frac{1}{L_F}$ be the learning rate. After T communication rounds, FedSSA satisfies*

$$\|\mathbf{w}^T - \mathbf{w}^*\|_2 \leq \left(1 - \frac{\lambda_F}{L_F + \lambda_F}\right)^T \|\mathbf{w}^0 - \mathbf{w}^*\|_2 + \frac{L_F + \lambda_F}{\lambda_F L_F} \mathcal{E}, \tag{17}$$

where $\mathbf{w}^*$ denotes the local optimal solution of $F(\mathbf{w})$, $F(\mathbf{w})$ denotes population risk (i.e., expected loss), L_F and λ_F denote the smoothness and strong convexity constants of $F(\mathbf{w})$, and $\mathcal{E}$ is aggregated error floor:

$$\mathcal{E} = \underbrace{C_1\left(\delta_\mu + \delta_\mu^2 + \delta_\Sigma\right)}_{\textit{semantic error}} + \underbrace{C_2(K+1)\epsilon_U + \lambda_1 C_3 + \lambda_2 C_4}_{\textit{structural error}}, \tag{18}$$

with constants $C_1, C_2, C_3, C_4 > 0$. Here δ_μ and δ_Σ represent bounds on the discrepancies of mean and covariance, while ϵ_U denotes the bound on Chordal distance.

The proof of Theorem 4.2 is provided in Appendix D. Moreover, we can derive the following corollary on the convergence rate from Eq. (17).

Corollary 4.3 (Convergence Rate)**.** *Under the conditions of Theorem 4.2, for any $\xi > 0$, if $T \geq \frac{L_F + \lambda_F}{\lambda_F} \log \frac{\|\mathbf{w}^0 - \mathbf{w}^*\|_2}{\xi}$, then the following holds:*

$$\|\mathbf{w}^T - \mathbf{w}^*\|_2 \leq \xi + \frac{L_F + \lambda_F}{\lambda_F L_F} \mathcal{E}. \tag{19}$$

Remark 4.4. Theorem 4.2 reveals that FedSSA converges at a linear rate to an $\mathcal{O}(\mathcal{E})$-neighborhood of the optimal solution $\mathbf{w}^*$. The error floor $\mathcal{E}$ consists of two components: (i) semantic error induced by aligning local distributions with cluster-level distributions, and (ii) structural error induced by aligning local spectral GNNs with cluster-level spectral GNNs. When clustering quality improves to the optimal (*i.e.*, $\delta_\mu, \delta_\Sigma, \epsilon_U \to 0$) and regularization parameters λ_1, λ_2 are sufficiently small, $\mathcal{E}$ approaches zero, and FedSSA converges to a neighborhood of the optimal solution.

5. Experiments

To validate the effectiveness of our FedSSA, we perform extensive experiments on eleven widely used datasets, which include both homophilic and heterophilic graphs. Specifically, we compare FedSSA with eleven baseline methods. Following previous work (Yu et al., 2025a), we evaluate 66 scenarios by varying the number of clients and adopting both non-overlapping and overlapping partitioning settings. To ensure a fair comparison, we compute the mean accuracy and standard deviation over ten independent runs. More detailed experimental settings can be found in Appendix F.

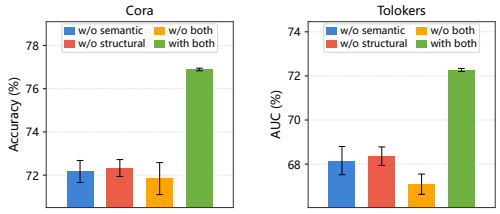
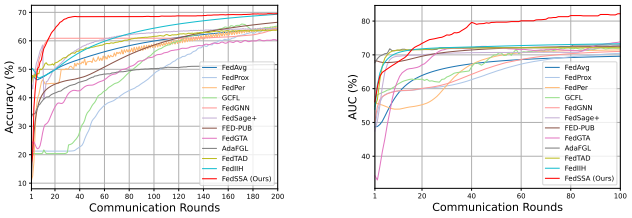
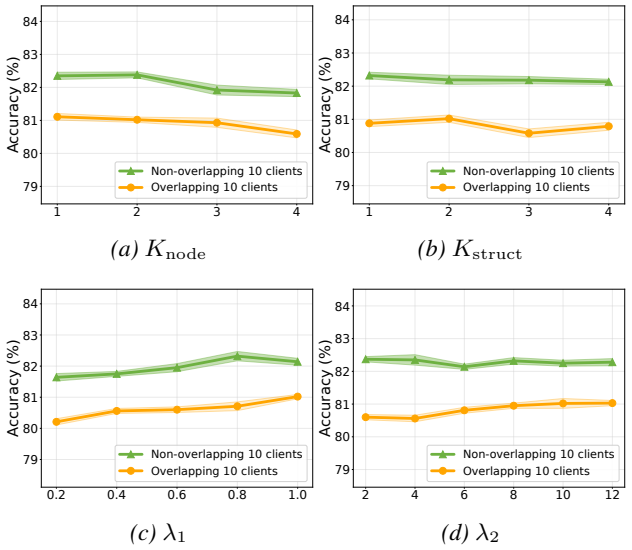


Figure 3. Ablation studies under overlapping partitioning setting with 30 clients.



(a) ogbn-arxiv *(b) Minesweeper*

Figure 4. Convergence curves on two datasets under non-overlapping partitioning setting with 10 clients.



(a) K_{node} *(b)* K_{struct} *(c)* λ_1 *(d)* λ_2

Figure 5. Accuracy curves with standard deviation bands on *Cora* dataset under different values of K_{node}, K_{struct}, λ_1, and λ_2.

5.1. Main Results

Homophilic Datasets Table 1 shows experimental results on homophilic datasets under non-overlapping partitioning setting. FedSSA achieves the best performance among all methods, and standard deviations are relatively small as well, which suggests that FedSSA is more effective and stable than the compared methods. Moreover, experimental results under overlapping partitioning setting are provided in Appendix H.1.

Heterophilic Datasets Table 2 shows experimental results on heterophilic datasets under non-overlapping partitioning setting. FedSSA not only achieves the best average performance among all baseline methods, but also outperforms the second-best method (*i.e.*, FedIIH) by 2.82% in terms of classification accuracy. This is because our FedSSA explicitly tackles both node feature heterogeneity and structural heterogeneity through semantic and structural knowledge alignment. Similarly, experimental results under overlapping partitioning setting are provided in Appendix H.1.

5.2. Ablation Study

To shed light on the contributions of components in our FedSSA, we perform ablation studies on *Cora* and *Tolokers* datasets, which are shown in Figure 3. Specifically, we employ 'w/o semantic' and 'w/o structural' to represent the reduced methods by removing 'sharing semantic knowledge' and 'sharing structural knowledge', respectively. It can be observed that the performance decreases when any component is removed, which demonstrates that each component contributes significantly. For example, the accuracies on *Cora* dataset are significantly decreased by more than 5% when both components are disabled. Ablation studies on other datasets are presented in Appendix H.2.

5.3. Convergence Curves

As shown in Figure 4, the convergence curves of our FedSSA exhibit small fluctuations, which validates its strong stability. Moreover, our FedSSA converges quickly within only a few communication rounds, which indicates its efficiency in real-world applications. For example, our method achieves convergence in an average of 60 communication rounds, while the compared baseline methods (*e.g.*, GCFL) require an average of 90 communication rounds. This superior performance can be attributed to our proposed semantic and structural knowledge sharing, which effectively mitigates the impact of heterogeneity among clients, thereby accelerating convergence. More convergence curves are provided in Appendix H.3.

5.4. Sensitivity Analysis on Hyperparameters

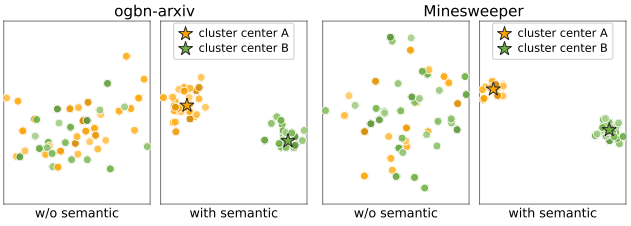
Here we perform a detailed sensitivity analysis of hyperparameters involved in our proposed FedSSA. Since our FedSSA includes four key hyperparameters (*i.e.*, the number of node clusters K_{node}, the number of structural clusters K_{struct}, and regularization hyperparameters λ_1 and λ_2), we plot accuracy curves with variance bars under different values of hyperparameters on *Cora* dataset. As shown in Figure 5, the performance variations under different values of hyperparameters are small, which validates that FedSSA is not sensitive to these hyperparameters. More sensitivity analyses are provided in Appendix H.4.

Table 1. Accuracy (%) of methods on six **homophilic** graph datasets under **non-overlapping** subgraph partitioning setting.

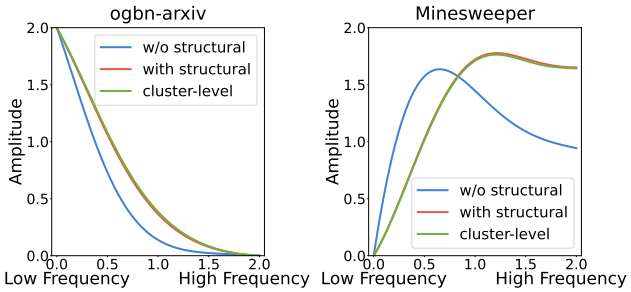
	Cora			CiteSeer			PubMed			-
Methods	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	-
Local	81.30±0.21	79.94±0.24	80.30±0.25	69.02±0.05	67.82±0.13	65.98±0.17	84.04±0.18	82.81±0.39	82.65±0.03	-
FedAvg (McMahan et al., 2017)	74.45±5.64	69.19±0.67	69.50±3.58	71.06±0.60	63.61±3.59	64.68±1.83	79.40±0.11	82.71±0.29	80.97±0.26	-
FedProx (Li et al., 2020)	72.03±4.56	60.18±7.04	48.22±6.18	71.73±1.11	63.33±3.25	64.85±1.35	79.45±0.25	82.55±0.24	80.50±0.25	-
FedPer (Arivazhagan et al., 2019)	81.68±0.40	79.35±0.04	78.01±0.32	70.41±0.32	70.53±0.28	66.64±0.27	85.80±0.21	84.20±0.28	84.72±0.31	-
GCFL (Xie et al., 2021)	81.47±0.65	78.66±0.27	79.21±0.70	70.34±0.57	69.01±0.12	66.33±0.05	85.14±0.33	84.18±0.19	83.94±0.36	-
FedGNN (Wu et al., 2021)	81.51±0.68	70.12±0.99	70.10±3.52	69.06±0.92	55.52±3.17	52.23±6.00	79.52±0.23	83.25±0.45	81.61±0.59	-
FedSage+(Zhang et al., 2021)	72.97±5.94	69.05±1.59	57.97±12.60	70.74±0.69	65.63±3.10	65.46±0.74	79.57±0.24	82.62±0.31	80.82±0.25	-
FED-PUB (Baek et al., 2023)	83.70±0.19	81.54±0.12	81.75±0.56	72.68±0.44	72.35±0.53	67.62±0.12	86.79±0.09	86.28±0.18	85.53±0.30	-
FedGTA (Li et al., 2023)	80.06±0.63	80.59±0.38	79.01±0.31	70.12±0.10	71.57±0.34	69.94±0.14	87.75±0.01	86.80±0.01	87.12±0.05	-
AdaFGL (Li et al., 2024)	82.01±0.51	80.09±0.08	79.74±0.05	71.44±0.27	72.34±0.09	70.95±0.45	86.91±0.28	86.97±0.10	86.59±0.21	-
FedTAD (Zhu et al., 2024)	80.31±0.26	80.87±0.11	80.07±0.15	70.34±0.37	69.43±0.75	68.09±0.69	84.00±0.13	84.61±0.17	84.33±0.18	-
FedIIH (Yu et al., 2025a)	84.11±0.17	81.85±0.09	83.01±0.15	72.86±0.25	76.50±0.06	73.36±0.41	87.80±0.18	87.65±0.18	87.19±0.25	-
FedSSA (Ours)	**84.67±0.05**	**82.32±0.04**	**84.13±0.09**	**73.06±0.08**	**77.65±0.10**	**74.09±0.09**	**88.11±0.07**	**87.78±0.13**	**87.37±0.14**	-
	Amazon-Computer			Amazon-Photo			ogbn-arxiv			Avg.
Methods	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	All
Local	89.22±0.13	88.91±0.17	89.52±0.20	91.67±0.09	91.80±0.02	90.47±0.15	66.76±0.07	64.92±0.09	65.06±0.05	79.57
FedAvg (McMahan et al., 2017)	84.88±1.96	79.54±0.23	74.79±0.24	89.89±0.83	83.15±3.71	81.35±1.04	65.54±0.07	64.44±0.10	63.24±0.13	74.58
FedProx (Li et al., 2020)	85.25±1.27	83.81±1.09	73.05±1.30	90.38±0.48	80.92±4.64	82.32±0.29	65.21±0.20	64.37±0.18	63.03±0.04	72.84
FedPer (Arivazhagan et al., 2019)	89.67±0.34	89.73±0.04	87.86±0.43	91.44±0.37	91.76±0.23	90.59±0.06	66.87±0.05	64.99±0.18	64.66±0.11	79.94
GCFL (Xie et al., 2021)	89.07±0.91	90.03±0.16	89.08±0.25	91.99±0.29	92.06±0.25	90.79±0.17	66.80±0.12	65.09±0.08	65.08±0.04	79.90
FedGNN (Wu et al., 2021)	88.08±0.15	88.18±0.41	83.16±0.13	90.25±0.70	87.12±2.01	81.00±4.48	65.47±0.22	64.21±0.32	63.80±0.05	75.23
FedSage+(Zhang et al., 2021)	85.04±0.61	80.50±1.13	70.42±0.85	90.77±0.44	76.81±8.24	80.58±1.15	65.69±0.09	64.52±0.14	63.31±0.20	73.47
FED-PUB (Baek et al., 2023)	90.74±0.05	90.55±0.13	90.12±0.09	93.29±0.19	92.73±0.18	91.92±0.12	67.77±0.09	66.58±0.08	66.64±0.12	81.59
FedGTA (Li et al., 2023)	86.69±0.18	86.66±0.23	85.01±0.87	93.33±0.12	93.50±0.21	92.61±0.15	60.32±0.04	60.22±0.09	58.74±0.14	79.45
AdaFGL (Li et al., 2024)	80.20±0.05	83.62±0.26	84.53±0.23	86.69±0.19	89.85±0.83	88.11±0.05	52.73±0.19	51.77±0.36	50.94±0.08	76.97
FedTAD (Zhu et al., 2024)	82.20±1.20	85.50±0.33	83.91±1.54	92.29±0.39	90.59±0.09	89.18±0.84	65.35±0.14	64.06±0.25	64.45±0.13	78.87
FedIIH (Yu et al., 2025a)	90.74±0.13	90.86±0.23	90.44±0.05	93.42±0.02	94.22±0.08	93.55±0.09	70.30±0.06	69.34±0.02	68.65±0.04	83.10
FedSSA (Ours)	**91.09±0.07**	**91.30±0.13**	**90.62±0.09**	**93.86±0.15**	**94.62±0.14**	**93.76±0.07**	**70.86±0.13**	**69.47±0.08**	**68.77±0.13**	**83.53**

5.5. Case Study

To illustrate the effectiveness of FedSSA in mitigating heterogeneity, we conduct case studies on two datasets. Specifically, we first visualize semantic representations of various clients obtained from 'w/o semantic' and 'with semantic' (*i.e.*, FedSSA without/with semantic knowledge sharing) by using t-SNE (Van der Maaten & Hinton, 2008). Subsequently, we plot spectral properties captured by local models under 'w/o structural' and 'with structural' (*i.e.*, FedSSA without/with structural knowledge sharing). As shown in Figure 6, the 2D projections of representations obtained from 'with semantic' show more compact clusters when compared with 'w/o semantic', which demonstrates the effectiveness of FedSSA in mitigating node feature heterogeneity. Moreover, spectral properties obtained from 'with structural' align more closely to those of cluster-level when compared with 'w/o structural', which validates the effectiveness of FedSSA in addressing structural heterogeneity. More case studies are provided in Appendix H.5.



(a) semantic representations of clients



(b) spectral properties of local models

Figure 6. Case studies on two datasets under overlapping partitioning setting with 50 clients.

6. Conclusion

In this paper, we propose a novel graph **Fed**erated learning method via **S**emantic and **S**tructural **A**lignment (FedSSA) to address heterogeneity in Graph Federated Learning (GFL). Specifically, instead of simply performing the weighted aggregation of model parameters, our proposed FedSSA enforces class-wise semantic knowledge sharing and structural knowledge sharing. On one hand, we minimize the divergence between local distributions and cluster-level dis-

Table 2. Comparisons on five **heterophilic** graph datasets under **non-overlapping** subgraph partitioning setting. Accuracy (%) is reported for *Roman-empire* and *Amazon-ratings*, and AUC (%) is reported for *Minesweeper*, *Tolokers*, and *Questions*.

	Roman-empire			Amazon-ratings			Minesweeper			-
Methods	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	-
Local	33.65±0.13	28.42±0.26	23.89±0.32	45.03±0.31	**45.89±0.19**	46.02±0.02	71.35±0.17	69.96±0.16	69.31±0.09	-
FedAvg (McMahan et al., 2017)	38.93±0.32	35.43±0.32	32.00±0.39	41.26±0.53	41.66±0.14	42.20±0.21	72.60±0.08	69.68±0.09	71.36±0.16	-
FedProx (Li et al., 2020)	27.95±0.59	26.43±1.41	23.12±0.49	36.92±0.02	36.86±0.14	36.96±0.05	71.91±0.27	70.66±0.20	71.50±0.37	-
FedPer (Arivazhagan et al., 2019)	20.75±1.75	15.51±1.13	15.45±2.76	36.62±0.30	32.34±1.01	36.96±0.03	58.73±10.45	65.35±7.02	53.80±11.40	-
GCFL (Xie et al., 2021)	40.65±0.14	40.51±0.24	37.85±0.25	36.92±0.05	36.86±0.14	36.96±0.02	72.04±0.13	71.88±0.12	69.20±0.18	-
FedGNN (Wu et al., 2021)	30.26±0.11	29.09±0.13	26.60±0.02	36.80±0.06	36.72±0.00	36.45±0.09	72.15±0.13	71.08±0.07	71.71±0.27	-
FedSage+(Zhang et al., 2021)	57.26±0.00	49.07±0.00	38.36±0.00	36.82±0.00	36.71±0.00	37.03±0.02	77.74±0.00	72.80±0.00	69.70±0.00	-
FED-PUB (Baek et al., 2023)	40.80±0.26	36.77±0.30	32.67±0.39	44.41±0.41	44.85±0.17	45.39±0.50	72.18±0.02	71.69±0.71	71.41±0.87	-
FedGTA (Li et al., 2023)	61.56±0.27	60.94±0.19	59.65±0.28	41.22±0.66	39.40±0.44	39.24±0.12	73.54±1.56	72.65±1.21	69.63±4.54	-
AdaFGL (Li et al., 2024)	67.64±0.18	64.55±0.09	62.42±0.26	41.70±0.06	42.30±0.00	42.59±0.14	73.24±1.13	70.79±2.14	71.26±1.31	-
FedTAD (Zhu et al., 2024)	45.26±0.19	44.71±0.38	42.04±0.13	43.59±0.33	43.35±0.29	44.50±0.26	72.39±0.43	71.99±0.13	72.74±0.03	-
FedIIH (Yu et al., 2025a)	68.32±0.05	66.44±0.28	64.61±0.13	44.26±0.24	44.24±0.10	45.19±0.04	74.29±0.02	73.23±0.04	72.81±0.02	-
FedSSA (Ours)	**68.67±0.10**	**66.81±0.09**	**65.14±0.15**	**45.18±0.14**	45.11±0.15	**46.13±0.05**	**82.26±0.14**	**82.16±0.08**	**82.60±0.11**	-

	Tolokers			Questions			Avg.			
Methods	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	5 Clients	10 Clients	20 Clients	All
Local	67.81±0.17	70.04±0.23	62.34±0.67	66.73±0.57	57.96±0.10	60.00±0.21	56.91	54.45	52.31	54.56
FedAvg (McMahan et al., 2017)	60.74±0.31	54.73±0.50	56.36±0.39	65.68±0.23	58.91±0.22	60.33±0.15	55.84	52.08	52.45	53.46
FedProx (Li et al., 2020)	42.90±0.24	41.15±0.22	40.42±0.62	47.36±0.38	45.46±0.34	46.83±0.11	45.41	44.11	43.77	44.43
FedPer (Arivazhagan et al., 2019)	46.61±9.88	54.97±13.23	44.82±11.61	58.38±9.39	59.40±9.71	62.32±1.56	44.22	45.51	42.67	44.13
GCFL (Xie et al., 2021)	64.39±1.17	59.90±0.85	58.82±0.70	60.51±1.18	59.85±0.16	60.31±0.48	54.90	53.80	52.63	53.78
FedGNN (Wu et al., 2021)	43.10±0.27	41.57±0.07	40.70±0.74	47.55±0.02	45.65±0.12	47.39±0.13	45.97	44.82	44.57	45.12
FedSage+(Zhang et al., 2021)	75.06±0.00	71.31±0.00	69.73±0.00	64.95±0.00	65.06±0.00	59.33±0.00	62.37	58.99	54.83	58.73
FED-PUB (Baek et al., 2023)	70.88±0.58	72.46±0.68	65.26±0.59	67.71±3.99	59.64±0.52	62.48±2.92	59.20	57.08	55.44	57.24
FedGTA (Li et al., 2023)	60.83±0.45	55.18±1.20	57.89±1.61	65.56±1.91	58.29±1.57	61.70±0.35	60.54	57.29	57.62	58.49
AdaFGL (Li et al., 2024)	59.26±2.18	54.78±2.12	56.61±2.93	64.23±2.09	58.82±1.14	62.84±0.49	61.21	58.25	59.14	59.54
FedTAD (Zhu et al., 2024)	60.91±0.25	53.39±1.73	56.47±1.58	68.89±1.20	58.44±1.06	61.51±1.56	58.21	54.38	55.45	56.01
FedIIH (Yu et al., 2025a)	71.09±0.26	71.32±0.09	70.30±0.10	68.32±0.03	67.99±0.09	65.40±0.07	65.26	64.64	63.66	64.52
FedSSA (Ours)	**75.82±0.05**	**73.96±0.10**	**72.29±0.11**	**69.51±0.15**	**68.69±0.18**	**65.74±0.07**	**68.29**	**67.35**	**66.10**	**67.34**

tributions to mitigate the heterogeneity in node features. On the other hand, we align the spectral characteristics of local spectral GNNs with those of cluster-level spectral GNNs to address the heterogeneity in structural topologies. By separately addressing two types of heterogeneity, our proposed FedSSA achieves strong performance on eleven datasets and outperforms the second-best method by a large margin of 2.82% in terms of classification accuracy.

Acknowledgements

Chen Gong was supported by NSF of China (Nos. 62336003, 12371510). Shuo Chen was supported by National Major S&T Special Project on New Generation Artificial Intelligence (No. 2025ZD0123500), National Natural Science Fund of China (No. 62506155), Provincial Natural Science Fund of Jiangsu (No. BK20251985), and Suzhou Municipal Leading Talents Fund (No. ZXL2025320). Wentao Yu and Bo Han were supported by RGC General Research Fund (No. 12200725) and NSFC General Program (No. 62376235). Sheng Wan was supported by the National Natural Science Foundation of China (No. 62506171), the Natural Science Foundation of Jiangsu Province (No. BK20241469), and the Fundamental Research Funds for the Central Universities (No: YDZX2026052).

Impact Statement

This work advances the development of Graph Federated Learning (GFL) methods that satisfy both privacy preservation and model effectiveness, thereby improving the performance in distributed graph learning scenarios. We encourage continued research into legally compliant and reliable GFL approaches that respect individual rights and intellectual property while maintaining robustness across diverse applications. By facilitating the broader deployment of GFL methods that can adapt to evolving legal and ethical standards, this research contributes to ensuring that such technologies remain trustworthy and socially beneficial.

References

Arivazhagan, M. G., Aggarwal, V., Singh, A. K., and Choudhary, S. Federated learning with personalization layers. *arXiv:1912.00818*, pp. 1–13, 2019. URL https://arxiv.org/abs/1912.00818.

Baek, J., Jeong, W., Jin, J., Yoon, J., and Hwang, S. J. Personalized subgraph federated learning. In *International Conference on Machine Learning*, pp. 1396–1415, 2023.

Bai, J., Yu, W., Xiao, Z., Havyarimana, V., Regan, A. C., Jiang, H., and Jiao, L. Two-stream spatial–temporal graph convolutional networks for driver drowsiness detection. *IEEE Transactions on Cybernetics*, 52(12):13821–13833, 2022.

Carrillo, J. A., Trillos, N. G., Li, S., and Zhu, Y. FedCBO: Reaching Group Consensus in Clustered Federated Learning through Consensus-based Optimization. *Journal of Machine Learning Research*, 25(214):1–51, 2024.

Chen, J., Lei, R., and Wei, Z. PolyGCL: Graph contrastive learning via learnable spectral polynomial filters. In *International Conference on Learning Representations*, pp. 1–30, 2024.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pp. 1–9, 2016.

Ghosh, A., Chung, J., Yin, D., and Ramchandran, K. An efficient framework for clustered federated learning. *IEEE Transactions on Information Theory*, 68(12):8076–8091, 2022.

Guo, Y. and Wei, Z. Graph neural networks with learnable and optimal polynomial bases. In *International Conference on Machine Learning*, pp. 12077–12097, 2023.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pp. 1–11, 2017.

He, M., Wei, Z., Huang, Z., and Xu, H. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. In *Advances in Neural Information Processing Systems*, pp. 14239–14251, 2021.

Hu, M., Zhou, P., Yue, Z., Ling, Z., Huang, Y., Li, A., Liu, Y., Lian, X., and Chen, M. FedCross: Towards accurate federated learning via multi-model cross-aggregation. In *International Conference on Data Engineering*, pp. 2137–2150, 2024.

Huang, K., Wang, Y. G., Li, M., and Lio, P. How universal polynomial bases enhance spectral graph neural networks: Heterophily, over-smoothing, and over-squashing. In *International Conference on Machine Learning*, pp. 1–20, 2024a.

Huang, W., Ye, M., Shi, Z., Wan, G., Li, H., Du, B., and Yang, Q. Federated learning for generalization, robustness, fairness: A survey and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46 (12):9387–9406, 2024b.

Karypis, G. METIS: Unstructured graph partitioning and sparse matrix ordering system. *Technical Report*, 1997.

Kim, H., Kim, H., and De Veciana, G. Clustered federated learning via gradient-based partitioning. In *International Conference on Machine Learning*, pp. 1–57, 2024.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv:1312.6114*, pp. 1–14, 2013. URL https://arxiv.org/abs/1312.6114.

Kingma, D. P., Rezende, D. J., Mohamed, S., and Welling, M. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pp. 1–9, 2014.

Kipf, T. N. and Welling, M. Variational graph auto-encoders. *arXiv:1611.07308*, pp. 1–3, 2016. URL https://arxiv.org/abs/1611.07308.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, pp. 1–14, 2017.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. In *Annual Conference on Machine Learning and Systems*, pp. 429–450, 2020.

Li, X., Wu, Z., Zhang, W., Zhu, Y., Li, R.-H., and Wang, G. FedGTA: Topology-aware averaging for federated graph learning. In *International Conference on Very Large Databases*, pp. 41–50, 2023.

Li, X., Wu, Z., Zhang, W., Sun, H., Li, R.-H., and Wang, G. AdaFGL: A new paradigm for federated node classification with topology heterogeneity. In *International Conference on Data Engineering*, pp. 2517–2530, 2024.

Ma, J., Cui, P., Kuang, K., Wang, X., and Zhu, W. Disentangled graph convolutional networks. In *International Conference on Machine Learning*, pp. 4212–4221, 2019.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282, 2017.

Meng, L., Shao, Y., Yuan, L., Lai, L., Cheng, P., Li, X., Yu, W., Zhang, W., Lin, X., and Zhou, J. A survey of distributed graph algorithms on massive graphs. *ACM Computing Surveys*, 57(2):1–39, 2024.

Platonov, O., Kuznedelev, D., Babenko, A., and Prokhorenkova, L. Characterizing graph datasets for node classification: Homophily-heterophily dichotomy and beyond. In *Advances in Neural Information Processing Systems*, pp. 523–548, 2023a.

Platonov, O., Kuznedelev, D., Diskin, M., Babenko, A., and Prokhorenkova, L. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In *International Conference on Learning Representations*, pp. 1–15, 2023b.

Sattler, F., Müller, K.-R., and Samek, W. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8):3710–3722, 2021.

Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vandergheynst, P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.

Valdeira, P., Wang, S., and Chi, Y. Vertical federated learning with missing features during training and inference. In *International Conference on Learning Representations*, pp. 1–20, 2025.

Van der Maaten, L. and Hinton, G. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11): 2579–2605, 2008.

Vardhan, H., Ghosh, A., and Mazumdar, A. An improved federated clustering algorithm with model-based clustering. *Transactions on Machine Learning Research*, 2(1): 1–28, 2024.

Wu, C., Wu, F., Cao, Y., Huang, Y., and Xie, X. FedGNN: Federated graph neural network for privacy-preserving recommendation. *arXiv:2102.04925*, pp. 1–9, 2021. URL https://arxiv.org/abs/2102.04925.

Xie, H., Ma, J., Xiong, L., and Yang, C. Federated graph classification over non-iid graphs. In *Advances in Neural Information Processing Systems*, pp. 18839–18852, 2021.

Yang, L., Chen, X., Zhuo, J., Jin, D., Wang, C., Cao, X., Wang, Z., and Guo, Y. Disentangled graph spectral domain adaptation. In *International Conference on Machine Learning*, pp. 1–17, 2025.

Yin, D., Chen, Y., Kannan, R., and Bartlett, P. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pp. 5650–5659, 2018.

Yu, W. Homophily heterogeneity matters in graph federated learning: A spectrum sharing and complementing perspective. *arXiv:2502.13732*, pp. 1–15, 2025. URL https://arxiv.org/abs/2502.13732.

Yu, W., Wan, S., Li, G., Yang, J., and Gong, C. Hyperspectral image classification with contrastive graph convolutional network. *IEEE Transactions on Geoscience and Remote Sensing*, 61(1):1–15, 2023.

Yu, W., Chen, S., Tong, Y., Gu, T., and Gong, C. Modeling inter-intra heterogeneity for graph federated learning. In *AAAI Conference on Artificial Intelligence*, pp. 22236–22244, 2025a.

Yu, W., Gong, C., Han, B., Fan, L., and Yang, Q. Integrating commonality and individuality for graph federated learning: A graph spectrum perspective. *Authorea Preprints*, pp. 1–16, 2025b. URL https://www.techrxiv.org/doi/full/10.36227/techrxiv.175502607.78991557.

Yu, W., Chen, S., Gong, C., Han, B., Niu, G., and Sugiyama, M. Atom-motif contrastive transformer for molecular property prediction. *ACM Transactions on Intelligent Systems and Technology*, 1(1):1–28, 2026a.

Yu, W., Han, B., Yang, J., and Gong, C. Beyond rigid alignment: Graph federated learning via dual manifold calibration. *arXiv:2605.06260*, pp. 1–30, 2026b. URL https://arxiv.org/abs/2605.06260.

Zhang, K., Yang, C., Li, X., Sun, L., and Yiu, S. M. Subgraph federated learning with missing neighbor generation. In *Advances in Neural Information Processing Systems*, pp. 6671–6682, 2021.

Zhang, W., Yin, Z., Sheng, Z., Li, Y., Ouyang, W., Li, X., Tao, Y., Yang, Z., and Cui, B. Graph attention multi-layer perceptron. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4560–4570, 2022.

Zhou, H., Yu, W., Wan, S., Tong, Y., Gu, T., and Gong, C. Traffic pattern sharing for federated traffic flow prediction with personalization. In *International Conference on Data Mining*, pp. 639–648, 2024.

Zhou, H., Yu, W., Wan, S., Tong, Y., Gu, T., and Gong, C. FedTPS: traffic pattern sharing for personalized federated traffic flow prediction. *Knowledge and Information Systems*, 67(1):5873–5899, 2025.

Zhou, H., Yu, W., Wei, Y., Li, G., Xu, S., and Gong, C. Inter-client dependency recovery with hidden global components for federated traffic prediction. In *AAAI Conference on Artificial Intelligence*, pp. 28946–28954, 2026.

Zhu, Y., Li, X., Wu, Z., Wu, D., Hu, M., and Li, R.-H. FedTAD: Topology-aware data-free knowledge distillation for subgraph federated learning. In *International Joint Conference on Artificial Intelligence*, pp. 1–9, 2024.

Table 3. Key notations used throughout the paper.

Notation	Description
$\mathcal{M}$	Set of clients, $\mathcal{M} = \{1, 2, \cdots, M\}$.
$\mathcal{G}_m = \langle \mathcal{V}_m, \mathcal{E}_m \rangle$	Local graph on client m with nodes $\mathcal{V}_m$ and edges $\mathcal{E}_m$.
n_m, e_m	Number of nodes and edges on client m.
d, C	Feature dimension and number of classes.
$\mathbf{X}_m \in \mathbb{R}^{n_m \times d}$	Node feature matrix of client m.
$\mathbf{A}_m \in \mathbb{R}^{n_m \times n_m}$	Adjacency matrix of client m.
$\mathbf{Y}_m$	Label matrix of client m.
$\mathbf{L}_m$	Symmetric normalized Laplacian of client m.
$\mathbf{U}_m, \mathbf{\Lambda}_m$	Eigenvectors and eigenvalues of $\mathbf{L}_m$.
$\tilde{\mathbf{X}}_m$	Graph Fourier transform of $\mathbf{X}_m$.
K_{node}	Number of node-level clusters.
K_{struct}	Number of structural clusters.
T	Total number of communication rounds.
λ_1, λ_2	Hyperparameters for regularization weights.
K	Number of polynomial filter orders.

A. Notations

For convenience, key notations used throughout the paper are summarized in Table 3.

B. Derivation of Eq. (2)

The log marginal likelihood $\log p(\mathbf{X}_m, \mathbf{Y}_m)$ can be derived as

$$\begin{aligned}
\log p(\mathbf{X}_m, \mathbf{Y}_m) &= \log \int p(\mathbf{X}_m, \mathbf{Y}_m, \mathbf{Z}_m)\, d\mathbf{Z}_m \\
&= \log \int p(\mathbf{X}_m|\mathbf{Z}_m, \mathbf{Y}_m)\, p(\mathbf{Z}_m)\, p(\mathbf{Y}_m)\, d\mathbf{Z}_m \\
&= \log \int p(\mathbf{X}_m|\mathbf{Z}_m, \mathbf{Y}_m)\, p(\mathbf{Z}_m)\, p(\mathbf{Y}_m)\, \frac{q(\mathbf{Z}_m|\mathbf{X}_m, \mathbf{Y}_m)}{q(\mathbf{Z}_m|\mathbf{X}_m, \mathbf{Y}_m)}\, d\mathbf{Z}_m \\
&= \log \mathbb{E}_{\mathbf{Z}_m \sim q(\mathbf{Z}_m|\mathbf{X}_m, \mathbf{Y}_m)} \frac{p(\mathbf{X}_m|\mathbf{Z}_m, \mathbf{Y}_m)\, p(\mathbf{Z}_m)\, p(\mathbf{Y}_m)}{q(\mathbf{Z}_m|\mathbf{X}_m, \mathbf{Y}_m)} \\
&\geq \mathbb{E}_{\mathbf{Z}_m \sim q(\mathbf{Z}_m|\mathbf{X}_m, \mathbf{Y}_m)} \log \frac{p(\mathbf{X}_m|\mathbf{Z}_m, \mathbf{Y}_m)\, p(\mathbf{Z}_m)\, p(\mathbf{Y}_m)}{q(\mathbf{Z}_m|\mathbf{X}_m, \mathbf{Y}_m)} \\
&\triangleq \mathcal{L}_{\text{ELBO}}(\mathbf{X}_m, \mathbf{Y}_m),
\end{aligned} \tag{20}$$

where the inequality is obtained by Jensen's inequality. Afterwards, we can further derive the Evidence Lower BOund (ELBO) $\mathcal{L}_{\text{ELBO}}(\mathbf{X}_m, \mathbf{Y}_m)$ as

$$\begin{aligned}
\mathcal{L}_{\text{ELBO}}(\mathbf{X}_m, \mathbf{Y}_m) &= \mathbb{E}_{\mathbf{Z}_m \sim q(\mathbf{Z}_m|\mathbf{X}_m, \mathbf{Y}_m)} \log p(\mathbf{X}_m|\mathbf{Z}_m, \mathbf{Y}_m) + \mathbb{E}_{\mathbf{Z}_m \sim q(\mathbf{Z}_m|\mathbf{X}_m, \mathbf{Y}_m)} [\log p(\mathbf{Z}_m) + \log p(\mathbf{Y}_m)] \\
&\quad - \mathbb{E}_{\mathbf{Z}_m \sim q(\mathbf{Z}_m|\mathbf{X}_m, \mathbf{Y}_m)} \log q(\mathbf{Z}_m|\mathbf{X}_m, \mathbf{Y}_m), \\
&= \mathbb{E}_{\mathbf{Z}_m \sim q(\mathbf{Z}_m|\mathbf{X}_m, \mathbf{Y}_m)} \log p(\mathbf{X}_m|\mathbf{Z}_m, \mathbf{Y}_m) + \mathbb{E}_{\mathbf{Z}_m \sim q(\mathbf{Z}_m|\mathbf{X}_m, \mathbf{Y}_m)} \log p(\mathbf{Y}_m) \\
&\quad + \mathbb{E}_{\mathbf{Z}_m \sim q(\mathbf{Z}_m|\mathbf{X}_m, \mathbf{Y}_m)} \log p(\mathbf{Z}_m) - \mathbb{E}_{\mathbf{Z}_m \sim q(\mathbf{Z}_m|\mathbf{X}_m, \mathbf{Y}_m)} \log q(\mathbf{Z}_m|\mathbf{X}_m, \mathbf{Y}_m), \\
&= \mathbb{E}_{\mathbf{Z}_m \sim q(\mathbf{Z}_m|\mathbf{X}_m, \mathbf{Y}_m)} \log p(\mathbf{X}_m|\mathbf{Z}_m, \mathbf{Y}_m) + \log p(\mathbf{Y}_m) - D_{\text{KL}}\big(q(\mathbf{Z}_m|\mathbf{X}_m, \mathbf{Y}_m) \parallel p(\mathbf{Z}_m)\big).
\end{aligned} \tag{21}$$

Finally, we obtain Eq. (2) in the main paper.

C. Details of Variational Graph AutoEncoder

In this section, we provide the architectural details of Variational Graph AutoEncoder (VGAE), which is proposed to infer the class-conditional latent distribution $q(\mathbf{Z}_m^c)$ in Eq. (3).

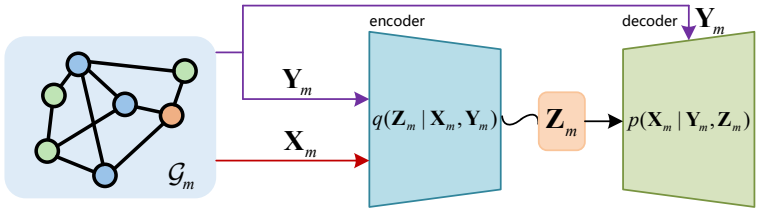


Figure 7. The architecture of our proposed VGAE, which comprises an encoder to infer latent variables and a decoder for graph reconstruction.

C.1. Architecture of VGAE

Here we present the architecture of our proposed VGAE. Since VGAE is deployed on each client, we take the m-th client as an illustrative example. Recall that the local graph $\mathcal{G}_m$ on client m consists of a node feature matrix $\mathbf{X}_m$ and the corresponding label matrix $\mathbf{Y}_m$. As illustrated in Figure 7, our VGAE is made up of an encoder and a decoder. The encoder approximates the posterior distribution $q(\mathbf{Z}_m|\mathbf{X}_m, \mathbf{Y}_m)$ by inferring latent variables $\mathbf{Z}_m$ from node features $\mathbf{X}_m$ conditioned on labels $\mathbf{Y}_m$. Specifically, we concatenate the one-hot encoding of node labels with node features as input to the encoder. Since class labels are explicitly incorporated, we can readily obtain the class-conditional distribution $q(\mathbf{Z}_m^c)$ for any given class c. Note that we employ reparameterization trick (Kingma & Welling, 2013) to sample $\mathbf{Z}_m$. Meanwhile, we employ a decoder to approximate the generative distribution $p(\mathbf{X}_m|\mathbf{Y}_m, \mathbf{Z}_m)$. Specifically, the decoder is implemented as an inner product layer following (Kipf & Welling, 2016).

C.2. ELBO Formulation

As ELBO is defined in Eq. (2), we provide its detailed formulation as follows. First, ELBO in Eq. (2) is composed of three terms. The first term (*i.e.*, $\mathbb{E}_{\mathbf{Z}_m \sim q(\mathbf{Z}_m|\mathbf{X}_m,\mathbf{Y}_m)} \log p(\mathbf{X}_m|\mathbf{Z}_m, \mathbf{Y}_m)$) is the expected log-likelihood, which can be implemented by a reconstruction loss. The second term (*i.e.*, $\log p(\mathbf{Y}_m)$) is the log prior of labels, which can be computed by the class distribution of labeled nodes. The third term (*i.e.*, $-D_{\mathrm{KL}}\big(q(\mathbf{Z}_m|\mathbf{X}_m, \mathbf{Y}_m) \parallel p(\mathbf{Z}_m)\big)$) is the negative KL divergence between the variational posterior and the prior of latent variables. Since we assume that both the variational posterior and prior follow Gaussian distributions, this term can be computed in closed form (Kingma & Welling, 2013). Finally, the overall ELBO can be optimized by maximizing these three terms.

D. Convergence Analysis of FedSSA

In this section, we first provide formal and detailed mathematical definitions for assumptions summarized in Assumption 4.1. Second, we provide the complete proof of Theorem 4.2, which consists of three parts: (i) error bound for semantic knowledge sharing, (ii) error bound for structural knowledge sharing, and (iii) overall convergence analysis.

D.1. Assumptions

To provide a rigorous theoretical foundation, we further decompose the summarized assumption in Assumption 4.1 into the following detailed assumptions, which are commonly adopted in the convergence analysis of federated learning (Hu et al., 2024; Yu et al., 2025b; Valdeira et al., 2025).

Assumption D.1 (Smoothness)**.** The population risk function $F(\mathbf{w})$ is L_F-smooth, namely, for all local models $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$,

$$\|\nabla F(\mathbf{w}) - \nabla F(\mathbf{w}')\|_2 \leq L_F \|\mathbf{w} - \mathbf{w}'\|_2.$$

Assumption D.2 (Strong Convexity)**.** The population risk function $F(\mathbf{w})$ is λ_F-strongly convex, namely, for all local models $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$,

$$F(\mathbf{w}') \geq F(\mathbf{w}) + \langle \nabla F(\mathbf{w}), \mathbf{w}' - \mathbf{w} \rangle + \frac{\lambda_F}{2} \|\mathbf{w}' - \mathbf{w}\|_2^2.$$

Assumption D.3 (Bounded Intra-cluster Heterogeneity)**.** For semantic knowledge sharing, the intra-cluster distribution divergence is bounded. Specifically, for any cluster $\mathcal{S}_i^c$ and any clients $m, n \in \mathcal{S}_i^c$, we assume

$$\|\boldsymbol{\mu}_m^c - \boldsymbol{\mu}_n^c\|_2 \leq \delta_\mu, \quad \|\boldsymbol{\Sigma}_m^c - \boldsymbol{\Sigma}_n^c\|_{\mathrm{F}} \leq \delta_\Sigma.$$

Moreover, let $\sigma_{\min}^2 \triangleq \min_{c\in\{1,2,\cdots,C\},\ i\in\{1,2,\cdots,K_{\text{node}}\}} \lambda_{\min}(\mathbf{\Sigma}_i^c)$ denote a uniform lower bound on the minimum eigenvalue of the cluster-level covariances $\mathbf{\Sigma}_i^c$ (defined in Eq. (6)), and assume $\sigma_{\min}^2 > 0$ and

$$\delta_\Sigma + \delta_\mu^2 \le \frac{\sigma_{\min}^2}{2}.$$

For structural knowledge sharing, the intra-cluster spectral energy divergence is bounded. Specifically, for any cluster $\mathcal{T}_j$ and any clients $m, n \in \mathcal{T}_j$, we assume

$$d_{\text{Chordal}}(\mathbf{Q}_m, \mathbf{Q}_n) \le \epsilon_U.$$

Assumption D.4 (Bounded Jacobians of Variational Parameters)**.** The variational parameters are differentiable functions of local model parameters $\mathbf{w} \in \mathcal{W}$. In particular, for each class c and client m, $q(\mathbf{Z}_m^c) = \mathcal{N}(\boldsymbol{\mu}_m^c(\mathbf{w}), \mathbf{\Sigma}_m^c(\mathbf{w}))$. We assume there exists a constant $L_q > 0$ such that for all $\mathbf{w} \in \mathcal{W}$,

$$\|\nabla_{\mathbf{w}} \boldsymbol{\mu}_m^c(\mathbf{w})\|_2 \le L_q, \qquad \|\nabla_{\mathbf{w}} \mathbf{\Sigma}_m^c(\mathbf{w})\|_{\text{F}} \le L_q,$$

where $\nabla_{\mathbf{w}} \boldsymbol{\mu}_m^c(\mathbf{w})$ denotes the Jacobian of $\boldsymbol{\mu}_m^c(\mathbf{w})$, and $\nabla_{\mathbf{w}} \mathbf{\Sigma}_m^c(\mathbf{w})$ denotes the Jacobian of $\mathbf{\Sigma}_m^c(\mathbf{w})$.

Assumption D.5 (Bounded and Lipschitz Spectral Coefficients)**.** The learnable coefficients of spectral GNNs satisfy the following properties.

- (Boundedness) For all $m \in \mathcal{M}$ and $k \in \{0, 1, \cdots, K\}$, there exists $w_{\max} > 0$ such that
$$|w_m^k| \le w_{\max}.$$
- (Lipschitz Dependence) There exists a constant $L_w > 0$ such that for any clients m, n and any filter order $k \in \{0, 1, \cdots, K\}$,
$$|w_m^k - w_n^k| \le L_w \, d_{\text{Chordal}}(\mathbf{Q}_m, \mathbf{Q}_n).$$

Lemma D.6 (Cluster-level Jacobian Bounds)**.** *Under Assumption D.3 and Assumption D.4, the cluster-level parameters* $\{\boldsymbol{\mu}_i^c(\mathbf{w}), \mathbf{\Sigma}_i^c(\mathbf{w})\}$ *defined in Eq.* (6) *are differentiable in* $\mathbf{w}$*. Moreover, for all* $\mathbf{w} \in \mathcal{W}$*,*

$$\|\nabla_{\mathbf{w}} \boldsymbol{\mu}_i^c(\mathbf{w})\|_2 \le L_q, \qquad \|\nabla_{\mathbf{w}} \mathbf{\Sigma}_i^c(\mathbf{w})\|_{\text{F}} \le (1 + 4\delta_\mu) L_q.$$

In particular, L_q *can be chosen sufficiently large so that the uniform bound* $\|\nabla_{\mathbf{w}} \mathbf{\Sigma}_i^c(\mathbf{w})\|_{\text{F}} \le L_q$ *holds.*

Proof. According to Eq. (6), the cluster-level mean is given by

$$\boldsymbol{\mu}_i^c(\mathbf{w}) = \sum_{n \in \mathcal{S}_i^c} \omega_n^c \boldsymbol{\mu}_n^c(\mathbf{w}).$$

Taking the Jacobian with respect to $\mathbf{w}$ and applying the triangle inequality, we obtain

$$\|\nabla_{\mathbf{w}} \boldsymbol{\mu}_i^c(\mathbf{w})\|_2 = \left\| \sum_{n \in \mathcal{S}_i^c} \omega_n^c \nabla_{\mathbf{w}} \boldsymbol{\mu}_n^c(\mathbf{w}) \right\|_2 \le \sum_{n \in \mathcal{S}_i^c} \omega_n^c \|\nabla_{\mathbf{w}} \boldsymbol{\mu}_n^c(\mathbf{w})\|_2 \le L_q.$$

In addition, for $\mathbf{\Sigma}_i^c(\mathbf{w})$ in Eq. (6), we have

$$\mathbf{\Sigma}_i^c(\mathbf{w}) = \sum_{n \in \mathcal{S}_i^c} \omega_n^c \mathbf{\Sigma}_n^c(\mathbf{w}) + \sum_{n \in \mathcal{S}_i^c} \omega_n^c \big(\boldsymbol{\mu}_n^c(\mathbf{w}) - \boldsymbol{\mu}_i^c(\mathbf{w})\big)\big(\boldsymbol{\mu}_n^c(\mathbf{w}) - \boldsymbol{\mu}_i^c(\mathbf{w})\big)^\top.$$

Taking the Jacobian with respect to $\mathbf{w}$ and applying the triangle inequality, we obtain

$$\|\nabla_{\mathbf{w}} \mathbf{\Sigma}_i^c(\mathbf{w})\|_{\text{F}} \le \sum_{n \in \mathcal{S}_i^c} \omega_n^c \|\nabla_{\mathbf{w}} \mathbf{\Sigma}_n^c(\mathbf{w})\|_{\text{F}} + \sum_{n \in \mathcal{S}_i^c} \omega_n^c 2 \|\boldsymbol{\mu}_n^c(\mathbf{w}) - \boldsymbol{\mu}_i^c(\mathbf{w})\|_2 \cdot \|\nabla_{\mathbf{w}} (\boldsymbol{\mu}_n^c(\mathbf{w}) - \boldsymbol{\mu}_i^c(\mathbf{w}))\|_2.$$

By Assumption D.3, $\|\boldsymbol{\mu}_n^c(\mathbf{w}) - \boldsymbol{\mu}_i^c(\mathbf{w})\|_2 \le \delta_\mu$, and from the previous result, $\|\nabla_{\mathbf{w}} \boldsymbol{\mu}_n^c(\mathbf{w})\|_2 \le L_q$, $\|\nabla_{\mathbf{w}} \boldsymbol{\mu}_i^c(\mathbf{w})\|_2 \le L_q$. Therefore, we have

$$\|\nabla_{\mathbf{w}} (\boldsymbol{\mu}_n^c(\mathbf{w}) - \boldsymbol{\mu}_i^c(\mathbf{w}))\|_2 \le \|\nabla_{\mathbf{w}} \boldsymbol{\mu}_n^c(\mathbf{w})\|_2 + \|\nabla_{\mathbf{w}} \boldsymbol{\mu}_i^c(\mathbf{w})\|_2 \le 2L_q.$$

Combining the above, we obtain

$$\|\nabla_{\mathbf{w}} \mathbf{\Sigma}_i^c(\mathbf{w})\|_{\text{F}} \le L_q + 2\delta_\mu \cdot 2L_q = (1 + 4\delta_\mu) L_q.$$

□

D.2. Error Bound for Semantic Knowledge Sharing

We first analyze the error introduced by aligning local distributions with cluster-level distributions. Recall that for each class c, the local distribution on client m is $q(\mathbf{Z}_m^c) = \mathcal{N}(\boldsymbol{\mu}_m^c, \boldsymbol{\Sigma}_m^c)$, and the cluster-level distribution for cluster $\mathcal{S}_i^c$ is $q(\mathcal{S}_i^c) = \mathcal{N}(\boldsymbol{\mu}_i^c, \boldsymbol{\Sigma}_i^c)$.

Lemma D.7 (KL Divergence Bound)**.** *Under Assumption D.3, for any client $m \in \mathcal{S}_i^c$, the KL divergence between the local distribution and the cluster-level distribution is bounded as*

$$D_{\mathrm{KL}}\big(q(\mathbf{Z}_m^c) \| q(\mathcal{S}_i^c)\big) \leq \frac{1}{2\sigma_{\min}^2}\delta_\mu^2 + \frac{3d}{2\sigma_{\min}^2}(\delta_\Sigma + \delta_\mu^2), \tag{22}$$

where $\sigma_{\min}^2 > 0$ is defined in Assumption D.3.

Proof. For two multivariate Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_m^c, \boldsymbol{\Sigma}_m^c)$ and $\mathcal{N}(\boldsymbol{\mu}_i^c, \boldsymbol{\Sigma}_i^c)$, the KL divergence admits the closed-form expression:

$$D_{\mathrm{KL}} = \frac{1}{2}\left(\mathrm{tr}((\boldsymbol{\Sigma}_i^c)^{-1}\boldsymbol{\Sigma}_m^c) + (\boldsymbol{\mu}_i^c - \boldsymbol{\mu}_m^c)^\top(\boldsymbol{\Sigma}_i^c)^{-1}(\boldsymbol{\mu}_i^c - \boldsymbol{\mu}_m^c) - d + \ln\frac{|\boldsymbol{\Sigma}_i^c|}{|\boldsymbol{\Sigma}_m^c|}\right). \tag{23}$$

We bound each term separately. For the first term, the cluster-level covariance $\boldsymbol{\Sigma}_i^c$ constructed in Eq. (6) can be rewritten as

$$\boldsymbol{\Sigma}_i^c = \sum_{n\in\mathcal{S}_i^c}\omega_n^c\boldsymbol{\Sigma}_n^c + \sum_{n\in\mathcal{S}_i^c}\omega_n^c(\boldsymbol{\mu}_n^c - \boldsymbol{\mu}_i^c)(\boldsymbol{\mu}_n^c - \boldsymbol{\mu}_i^c)^\top.$$

Therefore, using triangle inequality, Assumption D.3, and $\|\mathbf{v}\mathbf{v}^\top\|_{\mathrm{F}} = \|\mathbf{v}\|_2^2$, we have

$$\|\boldsymbol{\Sigma}_m^c - \boldsymbol{\Sigma}_i^c\|_{\mathrm{F}} \leq \underbrace{\left\|\boldsymbol{\Sigma}_m^c - \sum_{n\in\mathcal{S}_i^c}\omega_n^c\boldsymbol{\Sigma}_n^c\right\|_{\mathrm{F}}}_{\leq\delta_\Sigma} + \underbrace{\left\|\sum_{n\in\mathcal{S}_i^c}\omega_n^c(\boldsymbol{\mu}_n^c - \boldsymbol{\mu}_i^c)(\boldsymbol{\mu}_n^c - \boldsymbol{\mu}_i^c)^\top\right\|_{\mathrm{F}}}_{\leq\delta_\mu^2} \leq \delta_\Sigma + \delta_\mu^2.$$

Consequently, we can bound the first term as

$$\mathrm{tr}((\boldsymbol{\Sigma}_i^c)^{-1}\boldsymbol{\Sigma}_m^c) = \mathrm{tr}((\boldsymbol{\Sigma}_i^c)^{-1}(\boldsymbol{\Sigma}_m^c - \boldsymbol{\Sigma}_i^c)) + d \leq \|(\boldsymbol{\Sigma}_i^c)^{-1}\|_{\mathrm{F}} \cdot \|\boldsymbol{\Sigma}_m^c - \boldsymbol{\Sigma}_i^c\|_{\mathrm{F}} + d \leq \frac{\sqrt{d}}{\sigma_{\min}^2}(\delta_\Sigma + \delta_\mu^2) + d \leq \frac{d(\delta_\Sigma + \delta_\mu^2)}{\sigma_{\min}^2} + d.$$

For the second term, we first note that Assumption D.3 together with the definition of cluster-level mean in Eq. (6) implies

$$\|\boldsymbol{\mu}_m^c - \boldsymbol{\mu}_i^c\|_2 = \left\|\sum_{n\in\mathcal{S}_i^c}\omega_n^c(\boldsymbol{\mu}_m^c - \boldsymbol{\mu}_n^c)\right\|_2 \leq \sum_{n\in\mathcal{S}_i^c}\omega_n^c\|\boldsymbol{\mu}_m^c - \boldsymbol{\mu}_n^c\|_2 \leq \delta_\mu.$$

Therefore, we obtain

$$(\boldsymbol{\mu}_i^c - \boldsymbol{\mu}_m^c)^\top(\boldsymbol{\Sigma}_i^c)^{-1}(\boldsymbol{\mu}_i^c - \boldsymbol{\mu}_m^c) \leq \|(\boldsymbol{\Sigma}_i^c)^{-1}\|_2 \cdot \|\boldsymbol{\mu}_i^c - \boldsymbol{\mu}_m^c\|_2^2 = \frac{1}{\sigma_{\min}^2}\|\boldsymbol{\mu}_i^c - \boldsymbol{\mu}_m^c\|_2^2 \leq \frac{\delta_\mu^2}{\sigma_{\min}^2}.$$

For the third term, we have

$$\left|\ln\frac{|\boldsymbol{\Sigma}_i^c|}{|\boldsymbol{\Sigma}_m^c|}\right| = \left|-\ln\det\left((\boldsymbol{\Sigma}_i^c)^{-1}\boldsymbol{\Sigma}_m^c\right)\right| = \left|\ln\det\left(\mathbf{I} + (\boldsymbol{\Sigma}_i^c)^{-1}(\boldsymbol{\Sigma}_m^c - \boldsymbol{\Sigma}_i^c)\right)\right|.$$

Let $\boldsymbol{\Delta} := \boldsymbol{\Sigma}_m^c - \boldsymbol{\Sigma}_i^c$ and define the symmetric matrix $\mathbf{H} := (\boldsymbol{\Sigma}_i^c)^{-\frac{1}{2}}\,\boldsymbol{\Delta}\,(\boldsymbol{\Sigma}_i^c)^{-\frac{1}{2}}$. Afterwards, according to $\det(\mathbf{I} + \mathbf{A}\mathbf{B}) = \det(\mathbf{I} + \mathbf{B}\mathbf{A})$, we have

$$\det\left(\mathbf{I} + (\boldsymbol{\Sigma}_i^c)^{-1}\boldsymbol{\Delta}\right) = \det\left(\mathbf{I} + (\boldsymbol{\Sigma}_i^c)^{-\frac{1}{2}}\boldsymbol{\Delta}(\boldsymbol{\Sigma}_i^c)^{-\frac{1}{2}}\right) = \det(\mathbf{I} + \mathbf{H}).$$

Let $\{\lambda_j\}_{j=1}^d$ be the eigenvalues of $\mathbf{H}$. Subsequently, we have

$$|\ln\det(\mathbf{I}+\mathbf{H})| = \left|\sum_{j=1}^{d}\ln(1+\lambda_j)\right| \le \sum_{j=1}^{d}|\ln(1+\lambda_j)|.$$

Moreover, Assumption D.3 implies

$$\|\mathbf{H}\|_2 \le \|(\mathbf{\Sigma}_i^c)^{-\frac{1}{2}}\|_2^2\,\|\mathbf{\Delta}\|_2 = \|(\mathbf{\Sigma}_i^c)^{-1}\|_2\,\|\mathbf{\Delta}\|_2 \le \frac{1}{\sigma_{\min}^2}\|\mathbf{\Delta}\|_{\mathrm{F}} \le \frac{\delta_\Sigma+\delta_\mu^2}{\sigma_{\min}^2} \le \frac{1}{2},$$

and thus $|\lambda_j| \le \|\mathbf{H}\|_2 \le \frac{1}{2}$ for all j. Applying the inequality $|\ln(1+x)| \le 2|x|$ for $|x| \le \frac{1}{2}$, we obtain

$$|\ln\det(\mathbf{I}+\mathbf{H})| \le 2\sum_{j=1}^{d}|\lambda_j| \le 2d\,\|\mathbf{H}\|_2 \le \frac{2d(\delta_\Sigma+\delta_\mu^2)}{\sigma_{\min}^2}.$$

Substituting the above three bounds into Eq. (23) completes the proof. □

Lemma D.8 (Gradient Error from Semantic Alignment)**.** *The gradient error induced by semantic knowledge sharing in Eq. (7) is bounded as*

$$\|\nabla_{\mathbf{w}}\mathcal{L}_{\text{node}}\|_2 \le C_1\left(\delta_\mu+\delta_\mu^2+\delta_\Sigma\right), \tag{24}$$

where $C_1 = \frac{C\cdot K_{\text{node}}\cdot C'}{\sigma_{\min}^2}$ *with* $C'>0$ *being a constant.*

Proof. Recall that the semantic knowledge alignment loss is defined as

$$\mathcal{L}_{\text{node}} = \sum_{c=1}^{C}\sum_{i=1}^{K_{\text{node}}}\sum_{m\in\mathcal{S}_i^c} D_{\mathrm{KL}}\big(q(\mathbf{Z}_m^c)\|q(\mathcal{S}_i^c)\big).$$

Here, $q(\mathbf{Z}_m^c)=\mathcal{N}(\boldsymbol{\mu}_m^c(\mathbf{w}),\mathbf{\Sigma}_m^c(\mathbf{w}))$ and $q(\mathcal{S}_i^c)=\mathcal{N}(\boldsymbol{\mu}_i^c(\mathbf{w}),\mathbf{\Sigma}_i^c(\mathbf{w}))$, where the cluster-level parameters $\{\boldsymbol{\mu}_i^c,\mathbf{\Sigma}_i^c\}$ are defined in Eq. (6).

Taking the gradient with respect to $\mathbf{w}$, we have

$$\nabla_{\mathbf{w}}\mathcal{L}_{\text{node}} = \sum_{c=1}^{C}\sum_{i=1}^{K_{\text{node}}}\sum_{m\in\mathcal{S}_i^c}\nabla_{\mathbf{w}} D_{\mathrm{KL}}\big(q(\mathbf{Z}_m^c)\|q(\mathcal{S}_i^c)\big).$$

By using the triangle inequality, we have

$$\|\nabla_{\mathbf{w}}\mathcal{L}_{\text{node}}\|_2 \le \sum_{c=1}^{C}\sum_{i=1}^{K_{\text{node}}}\sum_{m\in\mathcal{S}_i^c}\left\|\nabla_{\mathbf{w}} D_{\mathrm{KL}}\big(q(\mathbf{Z}_m^c)\|q(\mathcal{S}_i^c)\big)\right\|_2.$$

Recall the closed-form KL divergence between Gaussians in Eq. (23). Differentiating each term and applying the chain rule, the gradient of $D_{\mathrm{KL}}\big(q(\mathbf{Z}_m^c)\|q(\mathcal{S}_i^c)\big)$ with respect to $\mathbf{w}$ is a linear combination of (i) $\nabla_{\mathbf{w}}\boldsymbol{\mu}_m^c$, $\nabla_{\mathbf{w}}\boldsymbol{\mu}_i^c$ and (ii) $\nabla_{\mathbf{w}}\mathbf{\Sigma}_m^c$, $\nabla_{\mathbf{w}}\mathbf{\Sigma}_i^c$, multiplied by matrices such as $(\mathbf{\Sigma}_i^c)^{-1}$ and $(\mathbf{\Sigma}_m^c)^{-1}$. We now make the above statement explicit by bounding the contribution of each term in Eq. (23). For brevity, we fix c,i,m and define $\boldsymbol{\mu}_m := \boldsymbol{\mu}_m^c(\mathbf{w})$, $\boldsymbol{\mu}_i := \boldsymbol{\mu}_i^c(\mathbf{w})$, $\mathbf{\Sigma}_m := \mathbf{\Sigma}_m^c(\mathbf{w})$, and $\mathbf{\Sigma}_i := \mathbf{\Sigma}_i^c(\mathbf{w})$. Let $\mathbf{d}_\mu := \boldsymbol{\mu}_i - \boldsymbol{\mu}_m$ and $\mathbf{\Delta} := \mathbf{\Sigma}_m - \mathbf{\Sigma}_i$.

Step 1 (Uniform Inverse Bounds). By Assumption D.3, we have $\lambda_{\min}(\mathbf{\Sigma}_i) \ge \sigma_{\min}^2$, which implies

$$\|\mathbf{\Sigma}_i^{-1}\|_2 \le \frac{1}{\sigma_{\min}^2}.$$

Furthermore, since $\|\mathbf{\Delta}\|_2 \le \|\mathbf{\Delta}\|_{\mathrm{F}} \le \delta_\Sigma+\delta_\mu^2 \le \frac{\sigma_{\min}^2}{2}$, Weyl's inequality gives

$$\lambda_{\min}(\mathbf{\Sigma}_m) \ge \lambda_{\min}(\mathbf{\Sigma}_i) - \|\mathbf{\Delta}\|_2 \ge \sigma_{\min}^2 - \frac{\sigma_{\min}^2}{2} = \frac{\sigma_{\min}^2}{2},$$

and thus

$$\|\mathbf{\Sigma}_m^{-1}\|_2 \leq \frac{2}{\sigma_{\min}^2}.$$

Step 2 (Differential Form of Gaussian KL). From Eq. (23), recall that the KL divergence between two Gaussians can be written as a sum of trace, quadratic, and log-determinant terms. To compute its gradient with respect to $\mathbf{w}$, we first write its total differential. Using the matrix calculus identities $\mathrm{d}\ln|\mathbf{A}| = \mathrm{tr}(\mathbf{A}^{-1}\mathrm{d}\mathbf{A})$ and $\mathrm{d}(\mathbf{A}^{-1}) = -\mathbf{A}^{-1}(\mathrm{d}\mathbf{A})\mathbf{A}^{-1}$, we obtain

$$\begin{aligned}2\,\mathrm{d}D_{\mathrm{KL}} = &\,\mathrm{tr}\left(\mathbf{\Sigma}_i^{-1}\,\mathrm{d}\mathbf{\Sigma}_m\right) - \mathrm{tr}\left(\mathbf{\Sigma}_i^{-1}(\mathrm{d}\mathbf{\Sigma}_i)\mathbf{\Sigma}_i^{-1}\mathbf{\Sigma}_m\right)\\ &+ 2\,\mathbf{d}_\mu^\top\mathbf{\Sigma}_i^{-1}(\mathrm{d}\boldsymbol{\mu}_i - \mathrm{d}\boldsymbol{\mu}_m) - \mathbf{d}_\mu^\top\mathbf{\Sigma}_i^{-1}(\mathrm{d}\mathbf{\Sigma}_i)\mathbf{\Sigma}_i^{-1}\mathbf{d}_\mu\\ &+ \mathrm{tr}\left(\mathbf{\Sigma}_i^{-1}\,\mathrm{d}\mathbf{\Sigma}_i\right) - \mathrm{tr}\left(\mathbf{\Sigma}_m^{-1}\,\mathrm{d}\mathbf{\Sigma}_m\right),\end{aligned}$$

where d denotes the total differential (*i.e.*, the first-order variation) induced by an infinitesimal perturbation $\mathrm{d}\mathbf{w}$. Next, we regroup terms involving $\mathrm{d}\mathbf{\Sigma}_m$ and $\mathrm{d}\mathbf{\Sigma}_i$ and then obtain

$$\begin{aligned}2\,\mathrm{d}D_{\mathrm{KL}} = &\,\mathrm{tr}\left(\left(\mathbf{\Sigma}_i^{-1} - \mathbf{\Sigma}_m^{-1}\right)\mathrm{d}\mathbf{\Sigma}_m\right)\\ &- \mathrm{tr}\left(\left(\mathbf{\Sigma}_i^{-1}\mathbf{\Delta}\mathbf{\Sigma}_i^{-1} + \mathbf{\Sigma}_i^{-1}\mathbf{d}_\mu\mathbf{d}_\mu^\top\mathbf{\Sigma}_i^{-1}\right)\mathrm{d}\mathbf{\Sigma}_i\right)\\ &+ 2\,\mathbf{d}_\mu^\top\mathbf{\Sigma}_i^{-1}(\mathrm{d}\boldsymbol{\mu}_i - \mathrm{d}\boldsymbol{\mu}_m),\end{aligned} \tag{25}$$

where we use the identity $\mathbf{\Sigma}_i^{-1} - \mathbf{\Sigma}_i^{-1}\mathbf{\Sigma}_m\mathbf{\Sigma}_i^{-1} = -\mathbf{\Sigma}_i^{-1}\mathbf{\Delta}\mathbf{\Sigma}_i^{-1}$ with $\mathbf{\Delta} = \mathbf{\Sigma}_m - \mathbf{\Sigma}_i$.

Step 3 (Chain Rule and Term-by-Term Bounds). For any unit direction $\mathbf{u}$ in the parameter space (*i.e.*, $\|\mathbf{u}\|_2 = 1$), set $\mathrm{d}\mathbf{w} = \mathbf{u}$. According to Assumption D.4, we have $\|\mathrm{d}\boldsymbol{\mu}_m\|_2 \leq L_q$ and $\|\mathrm{d}\mathbf{\Sigma}_m\|_{\mathrm{F}} \leq L_q$ for all clients. By Lemma D.6, the same bounds hold for cluster-level parameters, namely $\|\mathrm{d}\boldsymbol{\mu}_i\|_2 \leq L_q$ and $\|\mathrm{d}\mathbf{\Sigma}_i\|_{\mathrm{F}} \leq L_q$. In addition, according to Assumption D.3, $\|\mathbf{d}_\mu\|_2 \leq \delta_\mu$ and $\|\mathbf{\Delta}\|_{\mathrm{F}} \leq \delta_\Sigma + \delta_\mu^2$. Therefore, we can bound each term in Eq. (25) as follows.

(a) Mean-related term. By applying the Cauchy-Schwarz inequality, we have

$$\left|\mathbf{d}_\mu^\top\mathbf{\Sigma}_i^{-1}(\mathrm{d}\boldsymbol{\mu}_i - \mathrm{d}\boldsymbol{\mu}_m)\right| \leq \|\mathbf{\Sigma}_i^{-1}\|_2\,\|\mathbf{d}_\mu\|_2\,(\|\mathrm{d}\boldsymbol{\mu}_i\|_2 + \|\mathrm{d}\boldsymbol{\mu}_m\|_2) \leq \frac{2L_q}{\sigma_{\min}^2}\,\delta_\mu.$$

(b) $\mathrm{d}\mathbf{\Sigma}_m$-related term. Note that

$$\mathbf{\Sigma}_i^{-1} - \mathbf{\Sigma}_m^{-1} = \mathbf{\Sigma}_i^{-1}\mathbf{\Delta}\mathbf{\Sigma}_m^{-1},$$

so

$$\|\mathbf{\Sigma}_i^{-1} - \mathbf{\Sigma}_m^{-1}\|_2 \leq \|\mathbf{\Sigma}_i^{-1}\|_2\,\|\mathbf{\Delta}\|_2\,\|\mathbf{\Sigma}_m^{-1}\|_2 \leq \frac{2}{\sigma_{\min}^4}\,\|\mathbf{\Delta}\|_{\mathrm{F}}.$$

Using $|\,\mathrm{tr}(\mathbf{A}^\top\mathbf{B})| \leq \|\mathbf{A}\|_{\mathrm{F}}\|\mathbf{B}\|_{\mathrm{F}}$ and $\|\mathbf{A}\|_{\mathrm{F}} \leq \sqrt{d}\|\mathbf{A}\|_2$, we obtain

$$\left|\mathrm{tr}\left(\left(\mathbf{\Sigma}_i^{-1} - \mathbf{\Sigma}_m^{-1}\right)\mathrm{d}\mathbf{\Sigma}_m\right)\right| \leq \sqrt{d}\,\|\mathbf{\Sigma}_i^{-1} - \mathbf{\Sigma}_m^{-1}\|_2\,\|\mathrm{d}\mathbf{\Sigma}_m\|_{\mathrm{F}} \leq \frac{2\sqrt{d}\,L_q}{\sigma_{\min}^4}\,\|\mathbf{\Delta}\|_{\mathrm{F}}.$$

(c) $\mathrm{d}\mathbf{\Sigma}_i$-related term. Similarly,

$$\|\mathbf{\Sigma}_i^{-1}\mathbf{\Delta}\mathbf{\Sigma}_i^{-1}\|_2 \leq \|\mathbf{\Sigma}_i^{-1}\|_2^2\,\|\mathbf{\Delta}\|_2 \leq \frac{1}{\sigma_{\min}^4}\,\|\mathbf{\Delta}\|_{\mathrm{F}}, \qquad \|\mathbf{\Sigma}_i^{-1}\mathbf{d}_\mu\mathbf{d}_\mu^\top\mathbf{\Sigma}_i^{-1}\|_2 \leq \|\mathbf{\Sigma}_i^{-1}\|_2^2\,\|\mathbf{d}_\mu\|_2^2 \leq \frac{\delta_\mu^2}{\sigma_{\min}^4}.$$

Therefore,

$$\left|\mathrm{tr}\left(\left(\mathbf{\Sigma}_i^{-1}\mathbf{\Delta}\mathbf{\Sigma}_i^{-1} + \mathbf{\Sigma}_i^{-1}\mathbf{d}_\mu\mathbf{d}_\mu^\top\mathbf{\Sigma}_i^{-1}\right)\mathrm{d}\mathbf{\Sigma}_i\right)\right| \leq \frac{\sqrt{d}\,L_q}{\sigma_{\min}^4}\left(\|\mathbf{\Delta}\|_{\mathrm{F}} + \delta_\mu^2\right).$$

Combining (a)–(c) and using $\|\mathbf{\Delta}\|_{\mathrm{F}} \leq \delta_\Sigma + \delta_\mu^2$, we conclude that there exists a constant $C' > 0$ (absorbing L_q, $\sqrt{d}$, and $1/\sigma_{\min}^2$) such that

$$\left\|\nabla_{\mathbf{w}} D_{\mathrm{KL}}\big(q(\mathbf{Z}_m^c)\|q(\mathcal{S}_i^c)\big)\right\|_2 \leq \frac{C'}{\sigma_{\min}^2}\left(\delta_\mu + \delta_\mu^2 + \delta_\Sigma\right).$$

Finally, summing over all c, i, m yields Eq. (24), where the counting factors are absorbed into C_1. □

D.3. Error Bound for Structural Knowledge Sharing

We now analyze the error introduced by aligning the spectral characteristics of local spectral GNNs with those of cluster-level spectral GNNs.

Lemma D.9 (Lipschitz Constant of the Spectral Filter)**.** *For client m, consider the spectral GNN defined as*

$$\mathbf{P}_m = \sum_{k=0}^{K} w_m^k \, \mathbf{H}_m^k, \tag{26}$$

where $\mathbf{H}_m^k = (\mathbf{L}_m)^k \mathbf{X}_m$. Let $h_m(x) = \sum_{k=0}^{K} w_m^k x^k$ be the corresponding spectral filter for $x \in [0, 2]$. Therefore, h_m is Lipschitz continuous on $[0, 2]$ with Lipschitz constant

$$L_{\text{filter},m} \triangleq \sup_{x \in [0,2]} |h'_m(x)| \leq \sum_{k=1}^{K} k \, |w_m^k| \, 2^{k-1}. \tag{27}$$

Consequently, controlling the magnitudes of polynomial coefficients (e.g., via regularization in Eq. (16)) directly constrains the spectral Lipschitz constant of the learned filter.

Proof. For any $x \in [0, 2]$, we have $h'_m(x) = \sum_{k=1}^{K} k \, w_m^k x^{k-1}$. Thus

$$|h'_m(x)| \leq \sum_{k=1}^{K} k \, |w_m^k| \, |x|^{k-1} \leq \sum_{k=1}^{K} k \, |w_m^k| \, 2^{k-1}.$$

Taking the supremum over $x \in [0, 2]$ yields Eq. (27). □

Remark D.10. Lemma D.9 characterizes the sensitivity of the spectral filter $h_m(x)$ with respect to perturbations in the spectral variable $x \in [0, 2]$. In particular, bounding the polynomial coefficients controls $\sup_{x \in [0,2]} |h'_m(x)|$, which limits how rapidly the learned filter can vary over the spectrum.

Note that the bound in Eq. (27) grows exponentially with K due to the 2^{k-1} term. However, in practice, K is set to a moderate value (*e.g.*, $K = 6$). Therefore, in practice, the Lipschitz constant of the spectral filter is much smaller than the worst-case theoretical bound.

Lemma D.11 (Coefficient Perturbation Bound)**.** *Fix a client m and let*

$$h_m(x) = \sum_{k=0}^{K} w_m^k x^k, \qquad \bar{h}_m(x) = \sum_{k=0}^{K} \bar{w}_m^k x^k, \qquad x \in [0, 2],$$

where $\bar{w}_m^k$ is a perturbed (e.g., cluster-consensus) coefficient and $\Delta w_m^k \triangleq w_m^k - \bar{w}_m^k$. Therefore, for any $x \in [0, 2]$,

$$|h_m(x) - \bar{h}_m(x)| \leq \sum_{k=0}^{K} |\Delta w_m^k| \, |x|^k \leq \sum_{k=0}^{K} |\Delta w_m^k| \, 2^k. \tag{28}$$

Equivalently,

$$\sup_{x \in [0,2]} |h_m(x) - \bar{h}_m(x)| \leq \left(\sum_{k=0}^{K} 4^k \right)^{\frac{1}{2}} \|\Delta w_m\|_2 = \left(\frac{4^{K+1} - 1}{3} \right)^{\frac{1}{2}} \|\Delta w_m\|_2, \tag{29}$$

where $\Delta w_m = [\Delta w_m^0, \Delta w_m^1, \cdots, \Delta w_m^K]^\top$. Moreover, for the polynomial-filter spectral GNN output

$$\mathbf{P}_m(w) = \sum_{k=0}^{K} w_m^k \mathbf{H}_m^k, \qquad \mathbf{P}_m(\bar{w}) = \sum_{k=0}^{K} \bar{w}_m^k \mathbf{H}_m^k,$$

we have the deterministic bound

$$\|\mathbf{P}_m(w) - \mathbf{P}_m(\bar{w})\|_{\text{F}} \leq \sum_{k=0}^{K} |\Delta w_m^k| \, \|\mathbf{H}_m^k\|_{\text{F}}. \tag{30}$$

In particular, since $\mathbf{L}_m$ *is the symmetric normalized Laplacian matrix,* $\|\mathbf{L}_m\|_2 \le 2$ *and then* $\|\mathbf{H}_m^k\|_{\mathrm{F}} \le 2^k \|\mathbf{X}_m\|_{\mathrm{F}}$. *Therefore,*

$$\|\mathbf{P}_m(w) - \mathbf{P}_m(\bar{w})\|_{\mathrm{F}} \le \|\mathbf{X}_m\|_{\mathrm{F}} \left(\frac{4^{K+1}-1}{3}\right)^{\frac{1}{2}} \|\Delta w_m\|_2.$$

Proof. By linearity,

$$h_m(x) - \bar{h}_m(x) = \sum_{k=0}^{K} (w_m^k - \bar{w}_m^k) x^k = \sum_{k=0}^{K} \Delta w_m^k x^k.$$

Applying triangle inequality gives

$$|h_m(x) - \bar{h}_m(x)| \le \sum_{k=0}^{K} |\Delta w_m^k| \, |x|^k \le \sum_{k=0}^{K} |\Delta w_m^k| \, 2^k,$$

which yields Eq. (28). For Eq. (29), after applying Cauchy-Schwarz inequality, we have

$$\sum_{k=0}^{K} |\Delta w_m^k| \, 2^k \le \Big(\sum_{k=0}^{K} 4^k\Big)^{\frac{1}{2}} \Big(\sum_{k=0}^{K} (\Delta w_m^k)^2\Big)^{\frac{1}{2}}.$$

Finally,

$$\mathbf{P}_m(w) - \mathbf{P}_m(\bar{w}) = \sum_{k=0}^{K} (w_m^k - \bar{w}_m^k) \mathbf{H}_m^k = \sum_{k=0}^{K} \Delta w_m^k \, \mathbf{H}_m^k,$$

and triangle inequality gives Eq. (30). □

Remark D.12 (Role of Lemma D.11). Lemma D.11 provides a deterministic forward-stability guarantee for polynomial-filter spectral GNNs: a small perturbation of the spectral coefficients implies a controlled change of the induced spectral filter h_m and the resulting representation $\mathbf{P}_m$. This lemma is used to justify the interpretability of structural alignment: aligning the coefficients stabilizes the learned spectral responses across clients, which supports structural knowledge sharing.

Lemma D.13 (Gradient Error from Structural Alignment)**.** *Let* $\overline{w}_j^k$ *denote the cluster mean coefficient defined in Eq.* (14). *Let* $c(m) \in \{1, 2, \cdots, K_{\mathrm{struct}}\}$ *be the structural cluster index such that* $m \in \mathcal{T}_{c(m)}$. *Define the cluster-consensus coefficient vector* w^{cm} *by*

$$(w^{\mathrm{cm}})_m^k \triangleq \overline{w}_{c(m)}^k, \qquad m = 1, 2, \cdots, M, \;\; k = 0, 1, \cdots, K.$$

Here w^{cm} *is a notational device for analysis (an 'instantaneous' within-cluster mean at the same iterate), rather than an explicit parameter substitution in the algorithm. Let* w *denote the collection of all spectral GNN coefficients* w_m^k *for all* $m = 1, 2, \cdots, M$ *and* $k = 0, 1, \cdots, K$.

Gradient convention. *In this lemma,* $\nabla F(\cdot)$ *denotes the gradient of the population risk restricted to the coefficient coordinates (all other model parameters are fixed). We reuse* L_F *to denote a smoothness constant on these coordinates.*

Define

$$e_{\mathrm{struct}}(w) \triangleq \nabla F(w^{\mathrm{cm}}) - \nabla F(w) + \lambda_1 \, g_{\ell_1}(w) + \lambda_2 \, w,$$

where $g_{\ell_1}(w) \in \partial \|w\|_1$ *denotes a subgradient of the* ℓ_1 *term in Eq.* (16) *(restricted to the coefficient coordinates). Therefore, we have*

$$\|e_{\mathrm{struct}}(w)\|_2 \le C_2(K+1)\epsilon_U + \lambda_1 C_3 + \lambda_2 C_4, \tag{31}$$

where ϵ_U *is defined in Assumption D.3.*

Proof. Recall that $\mathcal{L}_{\mathrm{struct}} = \mathcal{L}_{\mathrm{align}} + \mathcal{L}_{\mathrm{reg}}$ is optimized via alignment and regularization, and the bound concerns the induced perturbation term $e_{\mathrm{struct}}(w)$ defined above.

Step 1 (Alignment-induced Deviation). Fix any structural cluster $\mathcal{T}_j$ and any $m \in \mathcal{T}_j$. For each $k \in \{0, 1, \cdots, K\}$, by Assumption D.3 and Assumption D.5,

$$|w_m^k - \overline{w}_j^k| \le \frac{1}{|\mathcal{T}_j|} \sum_{n \in \mathcal{T}_j} |w_m^k - w_n^k| \le L_w \, \epsilon_U.$$

Therefore, for each $m \in \mathcal{T}_j$,

$$\|[w_m^0 - \overline{w}_j^0, w_m^1 - \overline{w}_j^1, \cdots, w_m^K - \overline{w}_j^K]\|_2 \le \|[w_m^0 - \overline{w}_j^0, w_m^1 - \overline{w}_j^1, \cdots, w_m^K - \overline{w}_j^K]\|_1 \le (K+1) L_w \, \epsilon_U,$$

and hence

$$\|w^{\mathrm{cm}} - w\|_2 \le \sqrt{M}\,(K+1) L_w \, \epsilon_U.$$

By L_F-smoothness (Assumption D.1), we have

$$\|\nabla F(w^{\mathrm{cm}}) - \nabla F(w)\|_2 \le L_F \, \|w^{\mathrm{cm}} - w\|_2 \le L_F \sqrt{M}\,(K+1) L_w \, \epsilon_U.$$

Step 2 (Regularization Terms). Since each coordinate of an ℓ_1 subgradient has magnitude at most 1, we have $\|g_{\ell_1}(w)\|_2 \le \sqrt{M(K+1)}$. By Assumption D.5, $\|w\|_2 \le \sqrt{M(K+1)}\, w_{\max}$ on the coefficient coordinates.

Step 3 (Combination). Setting

$$C_2 \triangleq L_F \sqrt{M}\, L_w, \quad C_3 \triangleq \sqrt{M(K+1)}, \quad C_4 \triangleq \sqrt{M(K+1)}\, w_{\max},$$

and applying the triangle inequality completes the proof. □

Remark D.14. The introduction of the cluster-consensus coefficients w^{cm} is a standard analytical device in clustered federated learning (Carrillo et al., 2024). Although the algorithm minimizes the structural alignment loss, the error bound is established by comparing the current coefficients to their cluster means. The actual structural alignment loss ensures that w_m^k remains close to $\overline{w}_j^k$, so the error induced by alignment is upper-bounded by the error at the consensus point.

D.4. Overall Convergence Analysis

We now prove Theorem 4.2 by combining the error bounds from semantic and structural knowledge sharing.

Proof of Theorem 4.2. Let $\mathbf{g}(\mathbf{w}^t)$ denote the local gradient used in FedSSA at iteration t, which incorporates both the task-specific gradient and the knowledge sharing gradients:

$$\mathbf{g}(\mathbf{w}^t) = \nabla F(\mathbf{w}^t) + \nabla_{\mathbf{w}} \mathcal{L}_{\mathrm{node}} + \mathbf{e}_{\mathrm{struct}}(\mathbf{w}^t).$$

By Lemma D.8 and Lemma D.13, the gradient error can be bounded as

$$\|\mathbf{g}(\mathbf{w}^t) - \nabla F(\mathbf{w}^t)\|_2 \le C_1(\delta_\mu + \delta_\mu^2 + \delta_\Sigma) + C_2(K+1)\epsilon_U + \lambda_1 C_3 + \lambda_2 C_4. \tag{32}$$

The update rule of FedSSA is given by $\mathbf{w}^{t+1} = \mathbf{w}^t - \eta \mathbf{g}(\mathbf{w}^t)$. To analyze the convergence, we consider the distance to the optimal solution:

$$\begin{aligned} \|\mathbf{w}^{t+1} - \mathbf{w}^*\|_2 &= \|\mathbf{w}^t - \eta \mathbf{g}(\mathbf{w}^t) - \mathbf{w}^*\|_2 \\ &\le \|\mathbf{w}^t - \eta \nabla F(\mathbf{w}^t) - \mathbf{w}^*\|_2 + \eta \|\mathbf{g}(\mathbf{w}^t) - \nabla F(\mathbf{w}^t)\|_2. \end{aligned} \tag{33}$$

For the first term, we use the co-coercivity property of strongly convex and smooth functions (Yin et al., 2018). By Assumption D.1 and Assumption D.2, for any $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$,

$$\langle \mathbf{w} - \mathbf{w}', \nabla F(\mathbf{w}) - \nabla F(\mathbf{w}') \rangle \ge \frac{L_F \lambda_F}{L_F + \lambda_F} \|\mathbf{w} - \mathbf{w}'\|_2^2 + \frac{1}{L_F + \lambda_F} \|\nabla F(\mathbf{w}) - \nabla F(\mathbf{w}')\|_2^2. \tag{34}$$

Setting $\mathbf{w}' = \mathbf{w}^*$ and noting that $\nabla F(\mathbf{w}^*) = \mathbf{0}$, we obtain

$$\langle \mathbf{w}^t - \mathbf{w}^*, \nabla F(\mathbf{w}^t) \rangle \ge \frac{L_F \lambda_F}{L_F + \lambda_F} \|\mathbf{w}^t - \mathbf{w}^*\|_2^2 + \frac{1}{L_F + \lambda_F} \|\nabla F(\mathbf{w}^t)\|_2^2. \tag{35}$$

We now bound $\|\mathbf{w}^t - \eta \nabla F(\mathbf{w}^t) - \mathbf{w}^*\|_2^2$ as

$$\begin{aligned} \|\mathbf{w}^t - \eta \nabla F(\mathbf{w}^t) - \mathbf{w}^*\|_2^2 &= \|\mathbf{w}^t - \mathbf{w}^*\|_2^2 - 2\eta \langle \mathbf{w}^t - \mathbf{w}^*, \nabla F(\mathbf{w}^t) \rangle + \eta^2 \|\nabla F(\mathbf{w}^t)\|_2^2 \\ &\le \|\mathbf{w}^t - \mathbf{w}^*\|_2^2 - \frac{2\eta L_F \lambda_F}{L_F + \lambda_F} \|\mathbf{w}^t - \mathbf{w}^*\|_2^2 \\ &\quad - \frac{2\eta}{L_F + \lambda_F} \|\nabla F(\mathbf{w}^t)\|_2^2 + \eta^2 \|\nabla F(\mathbf{w}^t)\|_2^2. \end{aligned} \tag{36}$$

Choosing $\eta = \frac{1}{L_F}$, the coefficient of $\|\nabla F(\mathbf{w}^t)\|_2^2$ simplifies to

$$-\frac{2}{L_F(L_F+\lambda_F)} + \frac{1}{L_F^2} = \frac{L_F+\lambda_F-2L_F}{L_F^2(L_F+\lambda_F)} = \frac{\lambda_F - L_F}{L_F^2(L_F+\lambda_F)} \leq 0,$$

since $\lambda_F \leq L_F$ for any smooth and strongly convex function. Therefore,

$$\|\mathbf{w}^t - \eta\nabla F(\mathbf{w}^t) - \mathbf{w}^*\|_2^2 \leq \left(1 - \frac{2\lambda_F}{L_F+\lambda_F}\right)\|\mathbf{w}^t - \mathbf{w}^*\|_2^2. \tag{37}$$

Using the inequality $\sqrt{1-x} \leq 1 - \frac{x}{2}$ for $x \in [0,1]$, we obtain

$$\|\mathbf{w}^t - \eta\nabla F(\mathbf{w}^t) - \mathbf{w}^*\|_2 \leq \sqrt{1 - \frac{2\lambda_F}{L_F+\lambda_F}}\|\mathbf{w}^t - \mathbf{w}^*\|_2 \leq \left(1 - \frac{\lambda_F}{L_F+\lambda_F}\right)\|\mathbf{w}^t - \mathbf{w}^*\|_2. \tag{38}$$

Substituting Eq. (32) and Eq. (38) into Eq. (33), we have

$$\|\mathbf{w}^{t+1} - \mathbf{w}^*\|_2 \leq \left(1 - \frac{\lambda_F}{L_F+\lambda_F}\right)\|\mathbf{w}^t - \mathbf{w}^*\|_2 + \frac{\mathcal{E}}{L_F}. \tag{39}$$

Let $\rho = 1 - \frac{\lambda_F}{L_F+\lambda_F} \in (0,1)$. Unrolling the recursion in Eq. (39) yields

$$\begin{aligned}
\|\mathbf{w}^T - \mathbf{w}^*\|_2 &\leq \rho^T\|\mathbf{w}^0 - \mathbf{w}^*\|_2 + \frac{\mathcal{E}}{L_F}\sum_{t=0}^{T-1}\rho^t \\
&\leq \rho^T\|\mathbf{w}^0 - \mathbf{w}^*\|_2 + \frac{\mathcal{E}}{L_F}\cdot\frac{1}{1-\rho} \\
&= \rho^T\|\mathbf{w}^0 - \mathbf{w}^*\|_2 + \frac{\mathcal{E}}{L_F}\cdot\frac{L_F+\lambda_F}{\lambda_F} \\
&= \left(1 - \frac{\lambda_F}{L_F+\lambda_F}\right)^T\|\mathbf{w}^0 - \mathbf{w}^*\|_2 + \frac{L_F+\lambda_F}{\lambda_F L_F}\mathcal{E}.
\end{aligned} \tag{40}$$

This completes the proof of Theorem 4.2. □

D.5. Proof of Corollary 4.3

Proof of Corollary 4.3. By Theorem 4.2, let

$$\rho \triangleq 1 - \frac{\lambda_F}{L_F+\lambda_F} \in (0,1).$$

Therefore, we have

$$\|\mathbf{w}^T - \mathbf{w}^*\|_2 \leq \rho^T\|\mathbf{w}^0 - \mathbf{w}^*\|_2 + \frac{L_F+\lambda_F}{\lambda_F L_F}\mathcal{E}.$$

If

$$T \geq \frac{L_F+\lambda_F}{\lambda_F}\log\frac{\|\mathbf{w}^0 - \mathbf{w}^*\|_2}{\xi},$$

then using $\log(1-x) \leq -x$ for $x \in (0,1)$, we have

$$\log\rho = \log\Big(1 - \frac{\lambda_F}{L_F+\lambda_F}\Big) \leq -\frac{\lambda_F}{L_F+\lambda_F}.$$

Let $a \triangleq \frac{\lambda_F}{L_F+\lambda_F}$. Therefore, we have $T\log\rho \leq -aT$. Moreover, since

$$T \geq \frac{1}{a}\log\frac{\|\mathbf{w}^0 - \mathbf{w}^*\|_2}{\xi},$$

multiplying both sides by $-a$ yields $-aT \leq \log \frac{\xi}{\|\mathbf{w}^0 - \mathbf{w}^*\|_2}$. Therefore,

$$T \log \rho \leq \log \frac{\xi}{\|\mathbf{w}^0 - \mathbf{w}^*\|_2}.$$

Exponentiating both sides yields

$$\rho^T \|\mathbf{w}^0 - \mathbf{w}^*\|_2 \leq \xi.$$

Substituting this into the bound from Theorem 4.2 gives

$$\|\mathbf{w}^T - \mathbf{w}^*\|_2 \leq \xi + \frac{L_F + \lambda_F}{\lambda_F L_F} \mathcal{E}.$$

□

D.6. Discussion on the Error Bound

Theorem 4.2 demonstrates that FedSSA converges linearly to a neighborhood of $\mathbf{w}^*$, whose radius is characterized by the error floor $\mathcal{E}$. In our bound, $\mathcal{E}$ consists of a semantic term and a structural term.

Semantic Term The term $C_1(\delta_\mu + \delta_\mu^2 + \delta_\Sigma)$ is induced by semantic alignment, namely matching local class-wise Gaussian posteriors to cluster-level distributions. This term is controlled by the intra-cluster heterogeneity parameters (*i.e.*, δ_μ and δ_Σ) in Assumption D.3, which quantifies the within-cluster discrepancy of class-wise means and covariances. Consequently, a tighter semantic clustering (*i.e.*, smaller within-cluster variation) leads to a smaller semantic error.

Structural Term The term $C_2(K+1)\epsilon_U + \lambda_1 C_3 + \lambda_2 C_4$ arises from structural alignment and regularization. The alignment component scales linearly with the cluster tightness ϵ_U and filter order (*i.e.*, $K+1$). The regularization components are linear in λ_1 and λ_2 under the coefficient boundedness condition in Assumption D.5. Therefore, decreasing ϵ_U, λ_1, and λ_2 reduces the structural error.

E. Efficiency Analysis

In this section, we analyze the space complexity and time complexity of our proposed FedSSA on client side and server side, respectively.

E.1. Space Complexity of Our Proposed FedSSA

E.1.1. Client Side

Since the local model deployed on each client is a spectral GNN (*i.e.*, UniFilter (Huang et al., 2024a)), the space complexity of our FedSSA on client side is determined by spectral GNN. Specifically, each local model consists of K orders of bases, and one Multi-Layer Perceptron (MLP). In other words, the space complexity of each client consists of two parts, namely K orders of bases and one MLP. On one hand, the space complexities of K bases are $\mathcal{O}(K \times n_m \times d)$. On the other hand, the space complexity of one MLP is $\mathcal{O}(d^2)$. Consequently, the overall space complexity of our method on client side is $\mathcal{O}(K \times n_m \times d + d^2)$.

E.1.2. Server Side

On server side, the space complexity of our method is determined by three parts, namely storing uploaded coefficients from M clients, storing inferred distributions from M clients, and storing spectral energy measures from M clients. First, the space complexity of storing uploaded coefficients from M clients is $\mathcal{O}(M \times K)$ when considering there are K orders of bases. Second, the space complexity of storing inferred distributions is $\mathcal{O}(M \times C \times d^2)$. Third, the space complexity of storing spectral energy measures is $\mathcal{O}(M \times K \times d)$. Consequently, the total space complexity of our method on server side is $\mathcal{O}(M \times K + M \times C \times d^2 + M \times K \times d)$. After simplification, it can be written as $\mathcal{O}\big(M \times (K + Cd^2 + Kd)\big)$.

Table 4. The space and time complexity of different methods on client side and server side. Here M, K, d, n_m, e_m, and C denote the number of clients, orders, dimensions, nodes, edges, and classes, respectively.

Method	Client Space	Server Space	Client Time	Server Time
FedAvg (McMahan et al., 2017)	$d + d^2$	$M(1 + d^2)$	$e_m d + n_m d^2$	M
FedProx (Li et al., 2020)	$d + 2d^2$	$M(1 + d^2)$	$e_m d + n_m d^2 + d^2$	M
FedPer (Arivazhagan et al., 2019)	$d + 2d^2$	$M(1 + d^2)$	$e_m d + n_m d^2 + d^2$	M
GCFL (Xie et al., 2021)	$d + d^2$	$M(1 + 2d^2)$	$e_m d + n_m d^2$	$M + M^2(\log M + d^2)$
FedGNN (Wu et al., 2021)	$d + 2d^2$	$2M(1 + 2d^2)$	$e_m d + n_m d^2 + d$	M
FedSage+(Zhang et al., 2021)	$n_m d + 3d^2$	$M(1 + 3d^2)$	$e_m d + n_m d^2$	M
FED-PUB (Baek et al., 2023)	$n_m d + d^2$	$M(d^2 + M)$	$e_m d + n_m d^2$	$Md(M + d)$
FedGTA (Li et al., 2023)	$d + d^2 + n_m C$	$M(1 + d^2 + n_m C)$	$e_m(d + n_m C) + n_m(d^2 + C)$	$M(1 + n_m C)$
AdaFGL (Li et al., 2024)	$n_m d + 2d^2$	$M(1 + d^2)$	$e_m d + e_m n_m + n_m d^2$	$Md + Mn_m d$
FedTAD (Zhu et al., 2024)	$n_m d + 2d^2$	$M(1 + d^2)$	$e_m d + n_m d^2$	$n_m d(d + n_m + 2MC)$
FedIIH (Yu et al., 2025a)	$Kn_m d + d^2$	$MK(d^2 + M)$	$K(e_m d + n_m d^2)$	$MKd(Mn_m + n_m d)$
FedSSA (Ours)	$Kn_m d + d^2$	$M(K + Cd^2 + Kd)$	$d(Ke_m + n_m d + C) + K$	$M(CK_{\text{node}} d + MdK + K_{\text{struct}})$

E.2. Time Complexity of Our Proposed FedSSA

E.2.1. Client Side

The time complexity of our FedSSA on each client is determined by our proposed VGAE, which is actually a spectral GNN. Specifically, it is made up of K orders of bases and one MLP. First, the time complexity of K orders of bases is $\mathcal{O}(K \times e_m \times d)$, where e_m denotes the number of edges. Second, the time complexity of one MLP is $\mathcal{O}(n_m \times d^2)$. Furthermore, the time complexities of semantic knowledge alignment and spectral knowledge alignment are $\mathcal{O}(C \times d)$ and $\mathcal{O}(K)$, respectively. Consequently, the overall time complexity of our method on client side is $\mathcal{O}(K \times e_m \times d + n_m \times d^2 + C \times d + K)$. After simplification, it can be written as $\mathcal{O}\big(d \times (Ke_m + n_m d + C) + K\big)$.

E.2.2. Server Side

On server side, there are two operations, namely clustering via inferred distributions (*a.k.a.*, semantic clustering) and clustering via spectral energies (*a.k.a.*, structural clustering). First, the time complexity of semantic clustering is $\mathcal{O}(M \times C \times K_{\text{node}} \times d)$. Second, the time complexity of structural clustering is $\mathcal{O}(M^2 dK + MK_{\text{struct}})$, where the first term accounts for computing all pairwise Chordal distances between spectral energy matrices, and the second term accounts for k-means clustering. Consequently, the total time complexity of our FedSSA on server side is $\mathcal{O}(M \times C \times K_{\text{node}} \times d + M(MdK + K_{\text{struct}}))$. After simplification, it can be written as $\mathcal{O}\big(M(CK_{\text{node}} d + MdK + K_{\text{struct}})\big)$.

E.3. Efficiency Comparison with Baseline Methods

As shown in Table 4, we present efficiency comparisons on the space complexity and time complexity of our proposed FedSSA with those of existing methods. We can observe that the complexities of our method are generally comparable to those of existing typical methods. This validates that our proposed FedSSA is efficient in real-world applications, which is confirmed by our experimental results in Section H.6.

F. Implementation Details

This section provides the details of our experimental setup, which includes local training objective, computational platform, graph datasets, subgraph partitioning, baseline methods, and training procedures.

F.1. Local Training Objective

For each client m, the overall local training objective integrates cross-entropy loss, negative ELBO loss, semantic alignment loss, and structural alignment loss. Specifically, we minimize

$$\mathcal{L}_m = \mathcal{L}_{\text{ce}}^{(m)} + \mathcal{L}_{\text{vgae}}^{(m)} + \mathcal{L}_{\text{node}}^{(m)} + \mathcal{L}_{\text{struct}}^{(m)}, \tag{41}$$

where $\mathcal{L}_{\text{ce}}^{(m)}$ is cross-entropy loss for node classification, $\mathcal{L}_{\text{vgae}}^{(m)}$ is negative ELBO loss (see Eq. (2)), $\mathcal{L}_{\text{node}}^{(m)}$ is semantic alignment loss, and $\mathcal{L}_{\text{struct}}^{(m)}$ is structural alignment loss.

Table 5. Statistical information of eleven used graph datasets.

Dataset	# Nodes	# Edges	# Classes	# Features
Homophilic Graphs				
Cora	2,708	5,429	7	1,433
CiteSeer	3,327	4,732	6	3,703
PubMed	19,717	44,324	3	500
Amazon-Computer	13,752	491,722	10	767
Amazon-Photo	7,650	238,162	8	745
ogbn-arxiv	169,343	1,166,243	40	128
Heterophilic Graphs				
Roman-empire	22,662	32,927	18	300
Amazon-ratings	24,492	93,050	5	300
Minesweeper	10,000	39,402	2	7
Tolokers	11,758	519,000	2	10
Questions	48,921	153,540	2	301

F.2. Experimental Platform

All experiments are conducted on a Linux server with a 2.90 GHz Intel Xeon Gold 6326 CPU, 64 GB RAM, and two NVIDIA GeForce RTX 4090 GPUs with 48 GB memory. Our FedSSA is implemented in Python 3.8.8, PyTorch 1.12.0, and PyTorch Geometric (PyG) 2.5.1.

F.3. Datasets

We evaluate our proposed FedSSA on eleven widely used benchmark datasets, which includes six homophilic and five heterophilic graph datasets. On one hand, homophilic graph datasets consist of four citation networks (*i.e.*, *Cora*, *CiteSeer*, *PubMed*, and *ogbn-arxiv*) and two Amazon product co-purchasing graphs (*i.e.*, *Amazon-Computer* and *Amazon-Photo*). On the other hand, heterophilic graph datasets include *Roman-empire*, *Amazon-ratings*, *Minesweeper*, *Tolokers*, and *Questions* (Platonov et al., 2023b). The statistical information of all datasets is summarized in Table 5. For *Minesweeper*, *Tolokers*, and *Questions*, which are binary classification tasks, we utilize Area Under the ROC Curve (AUC) as the evaluation metric following previous work (Platonov et al., 2023b; Yu et al., 2025a). For other multi-class datasets, classification accuracy is used as the evaluation metric.

For all datasets except *ogbn-arxiv*, we randomly sample 20% of nodes for training, 40% of nodes for validation, and 40% of nodes for testing. For *ogbn-arxiv*, which contains more than 0.1 million nodes and 1 million edges, we follow previous work (Baek et al., 2023; Yu et al., 2025a) and utilize only 5% of nodes for training, 47.5% of nodes for validation, and 47.5% of nodes for testing.

F.4. Subgraph Partitioning

Following practical scenarios and prior work (Baek et al., 2023; Yu et al., 2025a), we consider two subgraph partitioning schemes, namely non-overlapping and overlapping. In the non-overlapping setting, the global node set $\mathcal{V}$ is partitioned into M disjoint subsets $\{\mathcal{V}_m\}_{m=1}^{M}$ such that $\cup_{m=1}^{M}\mathcal{V}_m = \mathcal{V}$ and $\mathcal{V}_m \cap \mathcal{V}_n = \emptyset$ for $m \neq n$. Any partitioning scheme that does not satisfy this condition is referred to as overlapping. Specifically, procedures for both partitioning schemes are described below.

F.4.1. Non-overlapping Partitioning

Given M clients, we generate M non-overlapping subgraphs by applying METIS graph partitioning algorithm (Karypis, 1997) to the global graph. Each client is then assigned a unique subgraph corresponding to one partition output by METIS.

F.4.2. Overlapping Partitioning

Given M clients, we first employ METIS to partition the global graph into $\lfloor M/5 \rfloor$ subgraphs, where $\lfloor \cdot \rfloor$ denotes the floor function. For each generated subgraph, we randomly sample half of its nodes and their associated edges. Meanwhile, we repeat this sampling process five times to generate five distinct but overlapping subgraphs. Consequently, the total number of overlapping subgraphs matches the number of clients.

F.5. Baseline Methods

We compare our proposed FedSSA with eleven baseline methods, which includes one classic Federated Learning (FL) method (*i.e.*, FedAvg (McMahan et al., 2017)), two personalized FL methods (*i.e.*, FedProx (Li et al., 2020) and FedPer (Arivazhagan et al., 2019)), three general Graph Federated Learning (GFL) methods (*i.e.*, GCFL (Xie et al., 2021), FedGNN (Wu et al., 2021), and FedSage+ (Zhang et al., 2021)), and five personalized GFL approaches (*i.e.*, FED-PUB (Baek et al., 2023), FedGTA (Li et al., 2023), AdaFGL (Li et al., 2024), FedTAD (Zhu et al., 2024), and FedIIH (Yu et al., 2025a)). In addition, we introduce a local training method, where each client trains independently without federated aggregation. The details of these baseline methods are summarized as follows.

FedAvg (McMahan et al., 2017): A foundational FL method in which clients train local models independently and periodically send updates to a central server. The server aggregates the received parameters by averaging and broadcasts the aggregated global model back to all clients.

FedProx (Li et al., 2020): A personalized FL method that adds a proximal term to local objective, which penalizes the divergence between local model parameters and global model parameters. This regularization stabilizes local updates and enables clients to learn personalized models while leveraging global information.

FedPer (Arivazhagan et al., 2019): A personalized FL method that aggregates the backbone network parameters across clients during federated aggregation, while keeping the classification layer parameters personalized and updated locally on each client.

GCFL (Xie et al., 2021): A representative GFL method, which is originally designed for vertical GFL scenarios such as molecular property prediction (Yu et al., 2026a). Specifically, GCFL employs a bi-partitioning strategy that recursively divides clients into two disjoint groups based on the similarity of their gradients. Actually, this procedure is similar to clustered FL (Sattler et al., 2021). After that, model aggregation is performed only within each group.

FedGNN (Wu et al., 2021): A general GFL method that enhances local performance by exchanging node embeddings across clients. To be specific, when nodes in different clients have identical neighborhoods, FedGNN transfers corresponding node embeddings to expand local subgraphs, which leads to enriched local information via cross-client node embeddings.

FedSage+ (Zhang et al., 2021): A GFL baseline that reconstructs missing edges between subgraphs. Specifically, each client receives node representations from other clients and computes gradients based on the distance between local node features and received representations. These gradients are then sent back to other clients and used to train their corresponding neighbor generator, which facilitates the reconstruction of missing edges.

FED-PUB (Baek et al., 2023): A personalized GFL method that performs personalized model aggregation. It estimates inter-subgraph similarity by evaluating local model outputs on a test graph. Depending on these similarity scores, it carries out weighted aggregation of model parameters. Furthermore, each client learns a personalized sparse mask to select and update only partially aggregated parameters, which are relevant to its local subgraph.

FedGTA (Li et al., 2023): A personalized GFL method that estimates inter-subgraph similarity levels. Specifically, it consists of three steps. First, each client computes topology-aware local smoothing confidence and mixed moments of neighbor features. Second, in each communication round, these computed results are then uploaded to server along with local model parameters. Third, server performs weighted aggregation tailored to each client based on these estimated similarities.

AdaFGL (Li et al., 2024): A personalized GFL method that employs a decoupled two-step personalization strategy. In the first stage, standard federated training is conducted to obtain a global knowledge extractor via aggregation in the final round. In the second stage, each client performs personalized training on its local subgraph using extracted federated knowledge.

FedTAD (Zhu et al., 2024): A personalized GFL method that introduces a generator to synthesize pseudo graphs for data-free knowledge distillation. This approach enables effective knowledge transfer from local models to the global model, thereby mitigating the negative impact of heterogeneity.

FedIIH (Yu et al., 2025a): A personalized GFL method that simultaneously models both inter-heterogeneity and intra-heterogeneity. On one hand, inter-heterogeneity is captured from a multi-level global perspective via hierarchical variational inference, which facilitates accurate estimation of inter-subgraph similarity via graph data distributions. On the other hand, intra-heterogeneity is addressed by disentangling each subgraph into multiple latent factors, which allows fine-grained personalization.

Algorithm 1 FedSSA Client Algorithm

Input: Number of local training epochs E; number of classes C; the order of bases K; local subgraph $\mathcal{G}_m$; local node features $\mathbf{X}_m$; local labels $\mathbf{Y}_m$; received cluster-level representative distributions $\{\mathcal{N}(\boldsymbol{\mu}_i^c, \boldsymbol{\Sigma}_i^c)\}_{c=1}^C$ from server; received cluster-level coefficients $\{\overline{w}_j^k\}_{k=0}^K$ from server.
Output: Predicted labels for unlabeled nodes in local subgraph $\mathcal{G}_m$.
Download cluster-level representative distributions $\{\mathcal{N}(\boldsymbol{\mu}_i^c, \boldsymbol{\Sigma}_i^c)\}_{c=1}^C$ from server;
Download cluster-level coefficients $\{\overline{w}_j^k\}_{k=0}^K$ from server;
for each local epoch e from 1 **to** E **do**
  # Semantic Knowledge Alignment
  Employ VGAE to infer class-wise latent distributions $q(\mathbf{Z}_m^c) = \mathcal{N}(\boldsymbol{\mu}_m^c, \boldsymbol{\Sigma}_m^c)$ for each class c via Eq. (3), where $c = 1, 2, \cdots, C$;
  Compute semantic knowledge alignment loss $\mathcal{L}_{\text{node}}$ via Eq. (7);
  # Structural Knowledge Alignment
  Compute spectral energy measure $\mathbf{S}_m$ via Eq. (9) and Eq. (10);
  Compute structural knowledge sharing loss $\mathcal{L}_{\text{struct}}$ via Eq. (15) and Eq. (16);
  Update local model parameters by minimizing the overall local objective (see Appendix F.1 for details);
end for
Upload coefficients $\{w_m^k\}_{k=0}^K$ to server;
Upload inferred class-wise latent distributions $\{q(\mathbf{Z}_m^c)\}_{c=1}^C$ to server;
Upload computed spectral energy measure $\mathbf{S}_m$ to server;
Predict labels for unlabeled nodes in local subgraph $\mathcal{G}_m$.

Local: A non-federated baseline method in which each client trains its local model independently by only using local data. In this method, there is no federated aggregation or collaboration.

F.6. Training Details

For all baseline methods except FedSage+, FedGTA, FedIIH, and our proposed FedSSA, we employ a two-layer Graph Convolutional Network (GCN) (Kipf & Welling, 2017) followed by a linear classifier as the network architecture. Meanwhile, the hyperparameters of each baseline method are set according to the configurations in their original papers. For FedSage+, we adopt GraphSage (Hamilton et al., 2017) as the encoder and train a missing neighbor generator to mend missing edges among subgraphs. In addition, FedGTA employs a Graph Attention Multi-Layer Perceptron (GAMLP) (Zhang et al., 2022) as its backbone together with a linear classifier. Besides, for FedIIH, we utilize a node feature projection layer from DisenGCN (Ma et al., 2019) to extract node representations, which are then classified by a Multi-Layer Perceptron (MLP). In contrast, for our proposed FedSSA, we utilize a spectral GNN (*i.e.*, UniFilter (Huang et al., 2024a)) to extract node representations, which are subsequently fed into an MLP for node classification.

G. Pseudocode of Our Proposed FedSSA

In this section, we show the pseudocode of our proposed FedSSA for clients and server in Algorithm 1 and Algorithm 2, respectively.

H. Additional Experiments

In this section, we provide additional experiments. First, we provide additional experimental results on eleven datasets under overlapping partitioning setting. Second, we provide supplementary ablation studies on eleven datasets under both non-overlapping and overlapping partitioning settings. Third, we present additional convergence curves on other datasets under two representative settings, namely non-overlapping with 10 clients and overlapping with 30 clients. Fourth, we provide additional sensitivity analysis on hyperparameters. Fifth, we present additional case studies on other datasets to illustrate the effectiveness of our FedSSA in mitigating heterogeneity among clients. Finally, we analyze computational efficiency by reporting time consumption (seconds) per communication round.

Algorithm 2 FedSSA Server Algorithm

Input: Number of communication rounds R; number of clients M; number of classes C; the order of bases K; coefficients $\{w_m^k\}_{k=0}^K$ from client m; inferred class-wise latent distributions $\{q(\mathbf{Z}_m^c)\}_{c=1}^C$ from client m; spectral energy measure $\mathbf{S}_m$ from client m.
Output: Cluster-level representative distributions $\{\mathcal{N}(\boldsymbol{\mu}_i^c, \boldsymbol{\Sigma}_i^c)\}_{c=1}^C$; cluster-level coefficients $\{\overline{w}_j^k\}_{k=0}^K$.
for each communication round r from 1 **to** R **do**
  for client $m \in \{1, 2, \cdots, M\}$ **in parallel do**
    Perform Algorithm 1 on client m;
    Receive coefficients $\{w_m^k\}_{k=0}^K$ from client m;
    Receive inferred class-wise latent distributions $\{q(\mathbf{Z}_m^c) = \mathcal{N}(\boldsymbol{\mu}_m^c, \boldsymbol{\Sigma}_m^c)\}_{c=1}^C$ from client m;
    Receive computed spectral energy measure $\mathbf{S}_m$ from client m;
  end for
  # Semantic Clustering
  for each class c from 1 **to** C **do**
    Employ reparameterization trick to obtain $\tilde{\mathbf{Z}}_m^c$, where $m \in \mathcal{M}$, $\tilde{\mathbf{Z}}_m^c = \boldsymbol{\mu}_m^c + (\boldsymbol{\Sigma}_m^c)^{\frac{1}{2}} \boldsymbol{\epsilon}_m^c$;
    Cluster $\{\tilde{\mathbf{Z}}_m^c \mid m \in \mathcal{M}\}$ into K_{node} clusters to obtain $\{\mathcal{S}_i^c\}_{i=1}^{K_{\text{node}}}$ via Eq. (4);
    Construct cluster-level representative distributions $\{\mathcal{N}(\boldsymbol{\mu}_i^c, \boldsymbol{\Sigma}_i^c)\}_{i=1}^{K_{\text{node}}}$ via Eq. (6);
    for each client $m \in \mathcal{M}$ **do**
      Find i such that $m \in \mathcal{S}_i^c$ and record the semantic cluster index of client m as i;
      Send $\mathcal{N}(\boldsymbol{\mu}_i^c, \boldsymbol{\Sigma}_i^c)$ to client m;
    end for
  end for
  # Structural Clustering
  Compute orthonormal bases $\mathbf{Q}_m$ from $\mathbf{S}_m$ via QR decomposition, where $m \in \mathcal{M}$;
  Measure pairwise Chordal distances $d_{\text{Chordal}}(\mathbf{Q}_m, \mathbf{Q}_n)$ via Eq. (12);
  Cluster clients into K_{struct} clusters to obtain $\{\mathcal{T}_j\}_{j=1}^{K_{\text{struct}}}$ via Eq. (13);
  for each cluster j from 1 **to** K_{struct} **do**
    Compute cluster-level coefficients $\{\overline{w}_j^k\}_{k=0}^K$ via Eq. (14);
  end for
  for each client $m \in \mathcal{M}$ **do**
    Find j such that $m \in \mathcal{T}_j$ and record the structural cluster index of client m as j;
    Send $\{\overline{w}_j^k\}_{k=0}^K$ to client m.
  end for
end for

H.1. Additional Experiments on Overlapping Subgraph Partitioning Setting

Due to space limitations, we only present experimental results under non-overlapping partitioning setting in the main paper. Here, we provide additional experimental results on the same eleven datasets under overlapping partitioning setting in Table 6 and Table 7. We can observe that our proposed FedSSA consistently outperforms all baseline methods on all datasets with varying numbers of clients. For example, in Table 7, the average accuracy of FedSSA is 63.90%, which is 1.40% higher than the second-best method (*i.e.*, FedIIH). This further validates the effectiveness of our proposed FedSSA in handling graph data heterogeneity in GFL scenarios.

H.2. Additional Ablation Studies

To further evaluate the contribution of each key component in our proposed FedSSA, we carry out ablation studies on eleven datasets under both non-overlapping and overlapping partitioning settings. Specifically, since there are two essential components in our FedSSA (*i.e.*, sharing semantic knowledge and sharing structural knowledge), we employ 'Semantic' and 'Structural' to represent them, respectively. Based on the inclusion or exclusion of these components, we consider four different combinations, which are shown in Table 8. We can observe that performances consistently degrade across all datasets when any individual component is removed. This demonstrates that each component is essential and contributes significantly to the overall effectiveness of our proposed FedSSA.

Table 6. Accuracy (%) of methods on six **homophilic** graph datasets under **overlapping** subgraph partitioning setting.

	Cora			CiteSeer			PubMed			-
Methods	10 Clients	30 Clients	50 Clients	10 Clients	30 Clients	50 Clients	10 Clients	30 Clients	50 Clients	-
Local	73.98±0.25	71.65±0.12	76.63±0.10	65.12±0.08	64.54±0.42	66.68±0.44	82.32±0.07	80.72±0.16	80.54±0.11	-
FedAvg (McMahan et al., 2017)	76.48±0.36	53.99±0.98	53.99±4.53	69.48±0.15	66.15±0.64	66.51±1.00	82.67±0.11	82.05±0.12	80.24±0.35	-
FedProx (Li et al., 2020)	77.85±0.50	51.38±1.74	56.27±9.04	69.39±0.35	66.11±0.75	66.53±0.43	82.63±0.17	82.13±0.13	80.50±0.46	-
FedPer (Arivazhagan et al., 2019)	78.73±0.31	74.18±0.24	74.42±0.37	69.81±0.28	65.19±0.81	67.64±0.44	85.31±0.06	84.35±0.38	83.94±0.10	-
GCFL (Xie et al., 2021)	78.84±0.26	73.41±0.27	76.63±0.16	69.48±0.39	64.92±0.18	65.98±0.30	83.59±0.25	80.77±0.12	81.36±0.11	-
FedGNN (Wu et al., 2021)	70.63±0.83	61.38±2.33	56.91±0.82	68.72±0.39	59.98±1.52	58.98±0.98	84.25±0.07	82.02±0.22	81.85±0.10	-
FedSage+(Zhang et al., 2021)	77.52±0.46	51.99±0.42	55.48±11.5	68.75±0.48	65.97±0.02	65.93±0.30	82.77±0.08	82.14±0.11	80.31±0.68	-
FED-PUB (Baek et al., 2023)	79.60±0.12	75.40±0.54	77.84±0.23	70.58±0.20	68.33±0.45	69.21±0.30	85.70±0.08	85.16±0.10	84.84±0.12	-
FedGTA (Li et al., 2023)	76.42±0.62	75.63±0.33	77.69±0.14	70.43±0.08	71.71±0.33	69.19±0.32	85.34±0.42	84.99±0.05	84.47±0.06	-
AdaFGL (Li et al., 2024)	78.50±0.19	75.80±0.23	74.41±0.00	72.63±0.15	68.18±0.31	62.90±0.75	85.58±0.23	85.85±0.41	84.45±0.07	-
FedTAD (Zhu et al., 2024)	79.29±0.78	60.92±2.17	68.08±0.44	**73.47±0.16**	67.74±0.57	63.51±0.68	82.98±0.20	82.11±0.15	81.63±0.19	-
FedIIH (Yu et al., 2025a)	80.57±0.23	76.82±0.24	**78.58±0.25**	73.16±0.18	72.27±0.21	69.56±0.11	85.87±0.03	86.65±0.11	**85.65±0.12**	-
FedSSA (Ours)	**81.02±0.09**	**76.89±0.06**	**78.58±0.15**	72.19±0.08	**72.38±0.13**	**69.68±0.10**	**86.40±0.09**	**86.83±0.06**	85.20±0.04	-
	Amazon-Computer			Amazon-Photo			ogbn-arxiv			Avg.
Methods	10 Clients	30 Clients	50 Clients	10 Clients	30 Clients	50 Clients	10 Clients	30 Clients	50 Clients	All
Local	88.50±0.20	86.66±0.00	87.04±0.02	92.17±0.12	90.16±0.12	90.42±0.15	62.52±0.07	61.32±0.04	60.04±0.04	76.72
FedAvg (McMahan et al., 2017)	88.99±0.19	83.37±0.47	76.34±0.12	92.91±0.07	89.30±0.22	74.19±0.57	63.56±0.02	59.72±0.06	60.94±0.24	73.38
FedProx (Li et al., 2020)	88.84±0.20	83.84±0.89	76.60±0.47	92.67±0.19	89.17±0.40	72.36±2.06	63.52±0.11	59.86±0.16	61.12±0.04	73.38
FedPer (Arivazhagan et al., 2019)	89.30±0.04	87.99±0.23	88.22±0.27	92.88±0.24	91.23±0.16	90.92±0.38	63.97±0.08	62.29±0.04	61.24±0.11	78.42
GCFL (Xie et al., 2021)	89.01±0.22	87.24±0.09	87.02±0.22	92.45±0.10	90.58±0.11	90.54±0.08	63.24±0.02	61.66±0.10	60.32±0.01	77.61
FedGNN (Wu et al., 2021)	88.15±0.09	87.00±0.10	83.96±0.88	91.47±0.11	87.91±1.34	78.90±6.46	63.08±0.19	60.09±0.04	60.51±0.11	73.66
FedSage+(Zhang et al., 2021)	89.24±0.15	81.33±1.20	76.72±0.39	92.76±0.05	88.69±0.99	72.41±1.36	63.24±0.02	59.90±0.12	60.95±0.09	73.12
FED-PUB (Baek et al., 2023)	89.98±0.08	89.15±0.06	88.76±0.14	93.22±0.07	92.01±0.07	91.71±0.11	64.18±0.04	63.34±0.12	62.55±0.12	79.53
FedGTA (Li et al., 2023)	90.10±0.18	88.79±0.27	88.15±0.21	93.13±0.14	92.49±0.06	91.77±0.06	55.98±0.09	56.76±0.07	57.89±0.09	78.39
AdaFGL (Li et al., 2024)	80.49±0.00	80.42±0.00	82.12±0.00	89.24±0.00	88.34±0.00	87.68±0.00	56.81±0.06	55.17±0.00	54.82±0.00	75.74
FedTAD (Zhu et al., 2024)	79.09±5.63	79.48±0.85	77.05±0.07	81.94±3.09	86.58±1.75	84.38±1.33	58.45±0.15	57.75±0.54	56.52±0.14	73.39
FedIIH (Yu et al., 2025a)	90.15±0.04	89.56±0.19	**89.99±0.00**	93.38±0.00	94.17±0.04	93.25±0.16	66.69±0.09	66.10±0.03	65.67±0.06	81.01
FedSSA (Ours)	**90.41±0.07**	**89.65±0.12**	89.34±0.11	**93.72±0.15**	**94.44±0.11**	**93.36±0.14**	**67.44±0.12**	**66.38±0.13**	**65.73±0.07**	**81.09**

Furthermore, we conduct ablation studies to quantitatively evaluate the performance of employing the single Gaussian approximation and the GMM-based approximation on two datasets (*i.e.*, *Cora* and *Roman-empire*). Specifically, the GMM-based approximation models the distributions as a mixture of three Gaussian components. The results are summarized in Table 9. The cross (*i.e.*, $\times$) denotes that GMM-based approximation is not significantly better than single Gaussian approximation revealed by a paired t-test with significance level 0.05. We would like to clarify that the distributions approximated by our single Gaussian are not extremely diverse nor highly multimodal. First, our proposed method clusters clients based on their distributions before performing the approximation, which promotes homogeneity within each cluster and effectively mitigates distributional diversity. Second, given the increased similarity of distributions after clustering, a single Gaussian approximation is sufficient to capture the cluster-level characteristics, while being more computationally efficient than a GMM-based approximation. Third, as shown in Table 9, the performance differences between the single Gaussian approximation and the GMM-based approximation are only 0.11 and 0.20 on *Cora* and *Roman-empire*, respectively, which are negligible for real-world applications.

H.3. Additional Convergence Curves

To further evaluate the convergence of our proposed FedSSA and the compared methods, we present convergence curves on six datasets under non-overlapping partitioning setting (see Figure 8), and on eight datasets under overlapping partitioning setting (see Figure 9). We can observe that our proposed FedSSA converges stably. In contrast, the convergence curves of typical methods such as FedGTA (*e.g.*, Fig. 9(d)) exhibit pronounced instability. This instability arises because FedGTA relies on dynamically estimated similarity levels to guide federated aggregation. However, under strong client heterogeneity, local models can undergo substantial changes between communication rounds, leading to rapidly fluctuating similarity levels. These fluctuations, in turn, cause an inconsistent aggregation process and hinder stable knowledge transfer across clients. Consequently, the convergence process becomes more erratic, resulting in the observed oscillations in the convergence curves.

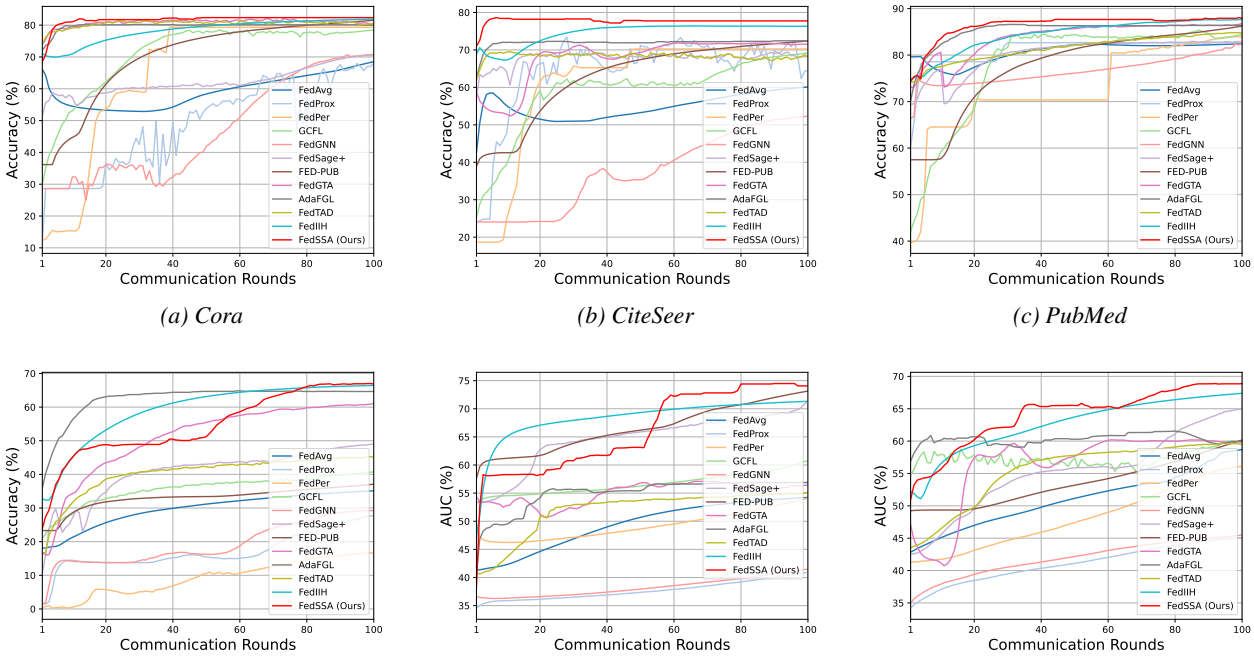
H.4. Additional Sensitivity Analysis on Hyperparameters

To further analyze the sensitivity of our proposed FedSSA to hyperparameters, we conduct additional sensitivity analyses on *Cora* and *Roman-empire* datasets. Specifically, we examine the impact of four key hyperparameters, namely the number of node clusters K_{node}, the number of structural clusters K_{struct}, and regularization parameters λ_1 and λ_2. Figure 10 and Figure 11 present accuracy curves with variance bars under different values of hyperparameters. Experimental results show

Table 7. Comparisons on five **heterophilic** graph datasets under **overlapping** subgraph partitioning setting. Accuracy (%) is reported for *Roman-empire* and *Amazon-ratings*, and AUC (%) is reported for *Minesweeper*, *Tolokers*, and *Questions*.

	Roman-empire			Amazon-ratings			Minesweeper			-
Methods	10 Clients	30 Clients	50 Clients	10 Clients	30 Clients	50 Clients	10 Clients	30 Clients	50 Clients	-
Local	39.47±0.03	34.43±0.14	31.28±0.18	41.43±0.04	41.81±0.14	42.57±0.12	67.98±0.07	64.39±0.10	62.73±0.23	-
FedAvg (McMahan et al., 2017)	40.89±0.25	38.66±0.08	36.71±0.20	39.86±0.06	41.40±0.02	41.02±0.16	69.06±0.07	67.95±0.04	66.89±0.08	-
FedProx (Li et al., 2020)	36.63±0.14	35.31±0.17	33.61±0.59	37.53±0.09	37.43±0.08	37.40±0.07	68.27±0.05	66.75±0.19	66.03±0.16	-
FedPer (Arivazhagan et al., 2019)	23.66±3.27	23.27±3.09	22.23±3.58	32.33±4.23	31.58±0.54	34.48±2.25	61.85±1.02	60.13±1.38	60.06±3.61	-
GCFL (Xie et al., 2021)	39.97±0.89	38.63±0.49	36.87±0.31	39.54±0.41	42.12±0.11	41.27±0.22	69.16±0.13	68.02±0.10	66.93±1.34	-
FedGNN (Wu et al., 2021)	37.46±0.12	36.47±0.24	34.92±0.26	36.58±0.16	36.77±0.12	36.95±0.15	68.59±0.21	67.30±0.17	66.41±0.23	-
FedSage+(Zhang et al., 2021)	57.48±0.00	42.55±0.00	37.13±0.00	36.86±0.00	36.71±0.00	37.03±0.00	**76.64±0.00**	70.56±0.00	70.34±0.00	-
FED-PUB (Baek et al., 2023)	43.80±0.25	40.46±0.16	37.73±0.09	42.25±0.25	42.30±0.06	42.88±0.34	69.11±0.13	67.76±0.24	67.52±0.14	-
FedGTA (Li et al., 2023)	59.86±0.04	58.32±0.09	57.57±0.21	40.81±0.24	39.44±0.06	39.37±0.04	70.64±0.40	67.99±1.60	67.20±1.35	-
AdaFGL (Li et al., 2024)	64.44±0.03	61.77±0.02	59.55±0.01	39.39±0.05	41.19±0.15	40.71±0.25	69.07±0.72	68.34±1.82	66.80±1.31	-
FedTAD (Zhu et al., 2024)	44.14±0.13	41.94±0.18	40.82±0.01	39.53±0.17	40.69±0.13	40.58±0.26	69.27±0.33	68.43±0.05	67.23±0.08	-
FedIIH (Yu et al., 2025a)	65.48±0.12	63.32±0.06	62.42±0.10	42.63±0.02	42.40±0.05	42.65±0.21	69.35±0.25	68.09±0.26	67.37±0.14	-
FedSSA (Ours)	**65.66±0.07**	**63.80±0.09**	**62.75±0.14**	**42.83±0.06**	**42.52±0.10**	**42.97±0.15**	75.65±0.11	**72.60±0.06**	**71.31±0.05**	-

	Tolokers			Questions			Avg.			
Methods	10 Clients	30 Clients	50 Clients	10 Clients	30 Clients	50 Clients	10 Clients	30 Clients	50 Clients	All
Local	73.83±0.03	69.01±0.31	66.63±0.20	63.17±0.02	57.17±0.08	56.13±0.02	57.18	53.36	51.87	54.14
FedAvg (McMahan et al., 2017)	72.99±0.40	58.51±0.27	55.47±0.42	62.80±0.63	58.88±0.18	60.78±0.27	57.12	53.08	52.17	54.12
FedProx (Li et al., 2020)	54.49±1.69	45.59±0.41	41.49±0.45	52.53±0.34	51.54±0.41	50.72±0.40	49.89	47.32	45.85	47.69
FedPer (Arivazhagan et al., 2019)	39.60±0.11	59.44±0.79	41.92±0.06	61.31±0.29	53.41±1.53	50.29±0.10	43.75	45.57	41.80	43.70
GCFL (Xie et al., 2021)	70.61±0.55	59.72±0.50	57.64±0.71	62.84±0.60	59.46±0.68	60.24±0.41	56.42	53.59	52.59	54.20
FedGNN (Wu et al., 2021)	56.21±1.20	46.85±0.31	42.18±0.45	53.25±0.15	51.90±0.15	51.22±0.14	50.42	47.86	46.34	48.20
FedSage+(Zhang et al., 2021)	**74.54±0.00**	70.88±0.00	69.61±0.00	64.22±0.00	65.34±0.00	62.76±0.00	61.95	57.21	55.37	58.18
FED-PUB (Baek et al., 2023)	74.17±0.29	70.35±0.54	66.80±0.85	65.39±2.44	58.38±1.19	60.73±0.74	58.94	55.85	55.13	56.64
FedGTA (Li et al., 2023)	70.34±1.53	59.59±1.10	56.12±1.48	63.20±1.27	58.11±1.06	60.99±0.77	60.97	56.69	56.25	57.97
AdaFGL (Li et al., 2024)	70.01±1.91	58.94±1.11	56.25±1.35	61.90±0.80	58.93±2.18	60.68±0.63	60.96	57.83	56.80	58.53
FedTAD (Zhu et al., 2024)	69.34±1.26	62.11±0.27	56.39±0.52	61.96±0.54	59.24±0.36	60.24±0.92	56.85	54.48	53.05	54.79
FedIIH (Yu et al., 2025a)	71.67±0.02	71.69±0.12	69.99±0.03	68.79±0.09	**66.98±0.04**	64.73±0.35	63.58	62.50	61.43	62.50
FedSSA (Ours)	74.33±0.09	**72.27±0.07**	**71.03±0.09**	**69.39±0.05**	66.43±0.13	**64.94±0.12**	**65.57**	**63.52**	**62.60**	**63.90**

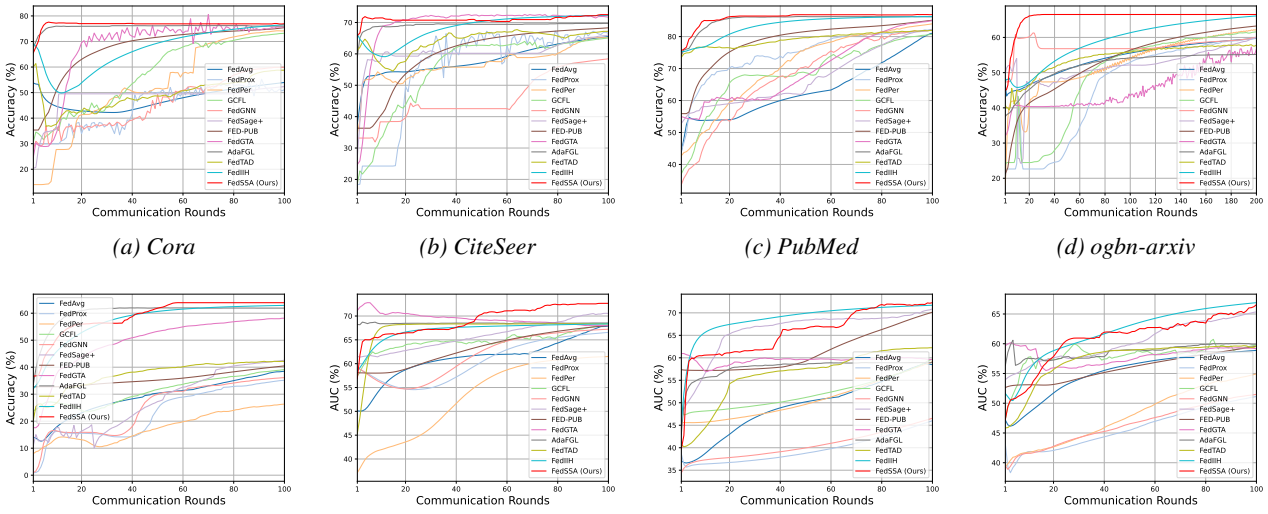


(a) Cora *(b) CiteSeer* *(c) PubMed*

(d) Roman-empire *(e) Tolokers* *(f) Questions*

Figure 8. Convergence curves on six datasets under non-overlapping partitioning setting with 10 clients.

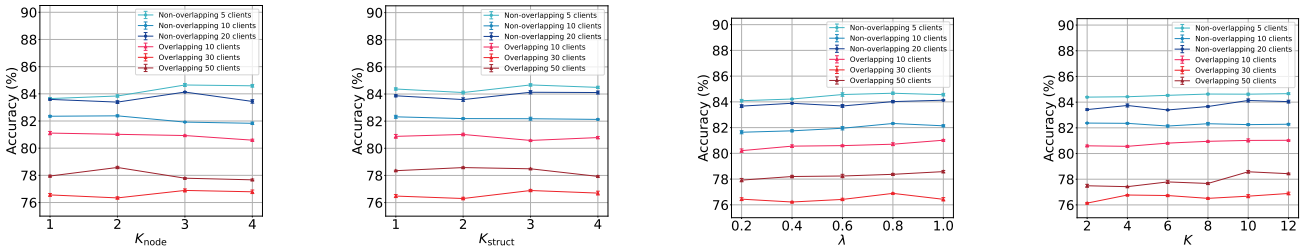
that our proposed FedSSA exhibits stable performance across a wide range of hyperparameters, which indicates that our FedSSA is not sensitive to the variation of hyperparameters.



(a) Cora *(b) CiteSeer* *(c) PubMed* *(d) ogbn-arxiv*

(e) Roman-empire *(f) Minesweeper* *(g) Tolokers* *(h) Questions*

Figure 9. Convergence curves on eight datasets under overlapping partitioning settings with 30 clients.



(a) Cora under different K_{node} *(b) Cora* under different K_{struct} *(c) Cora* under different λ_1 *(d) Cora* under different λ_2

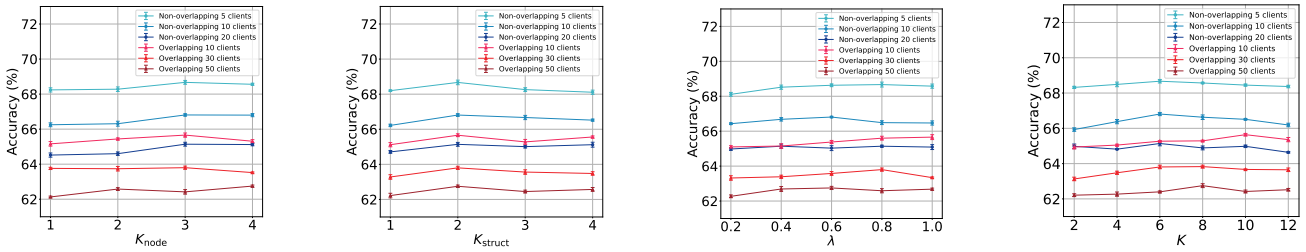
Figure 10. Accuracy curves with variance bars on *Cora* dataset under different values of K_{node}, K_{struct}, λ_1, and λ_2.

H.5. Additional Case Studies

To further illustrate the effectiveness of our proposed FedSSA in mitigating heterogeneity among clients, we conduct additional case studies on six datasets. To be specific, we first visualize semantic representations of various clients obtained from 'w/o semantic' and 'with semantic' (*i.e.*, FedSSA without/with semantic knowledge sharing) by using t-SNE (Van der Maaten & Hinton, 2008) method. Subsequently, we plot spectral properties captured by local models under 'w/o structural' and 'with structural' (*i.e.*, FedSSA without/with structural knowledge sharing). As shown in Figure 12, the 2D projections of different clients exhibit significant divergence under 'w/o semantic', which indicates the presence of heterogeneity among clients. In contrast, the 2D projections of representations obtained from 'with semantic' show more compact clusters when compared with 'w/o semantic', which demonstrates the effectiveness of FedSSA in mitigating node feature heterogeneity. Moreover, as shown in Figure 13, spectral properties obtained from 'with structural' align more closely to those of cluster-level when compared with 'w/o structural', which validates the effectiveness of our FedSSA in addressing structural heterogeneity.

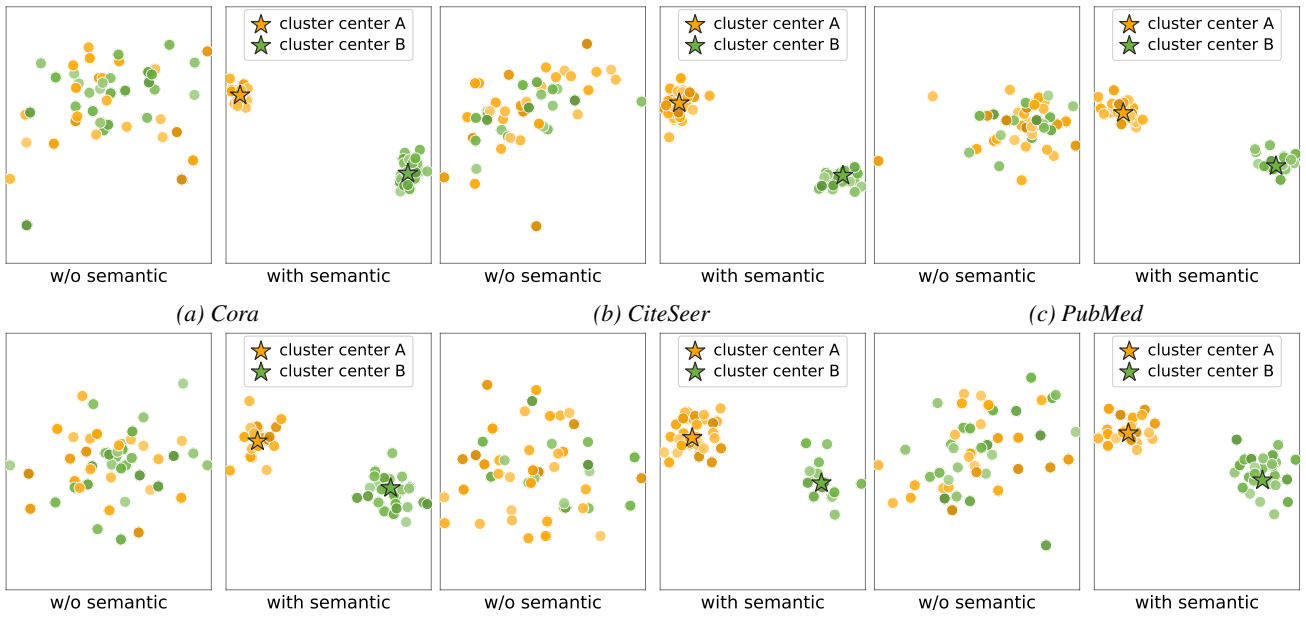
H.6. Time of Each Communication Round

We report the time consumed per communication round for our proposed FedSSA and baseline methods in Table 10. We can observe that FedSSA consistently achieves a lower time cost per communication round when compared with the average time of baseline methods. Notably, FedSSA is substantially more efficient than strong baseline methods such as FED-PUB and FedIIH. For example, on *Cora* dataset, FedSSA achieves more than a threefold speed improvement relative to the second-best baseline method (*i.e.*, FedIIH). This efficiency gain is attributed to the lower time complexity of FedSSA on



(a) Roman-empire under different K_{node} *(b) Roman-empire* under different K_{struct} *(c) Roman-empire* under different λ_1 *(d) Roman-empire* under different λ_2

Figure 11. Accuracy curves with variance bars on *Roman-empire* dataset under different values of K_{node}, K_{struct}, λ_1, and λ_2.

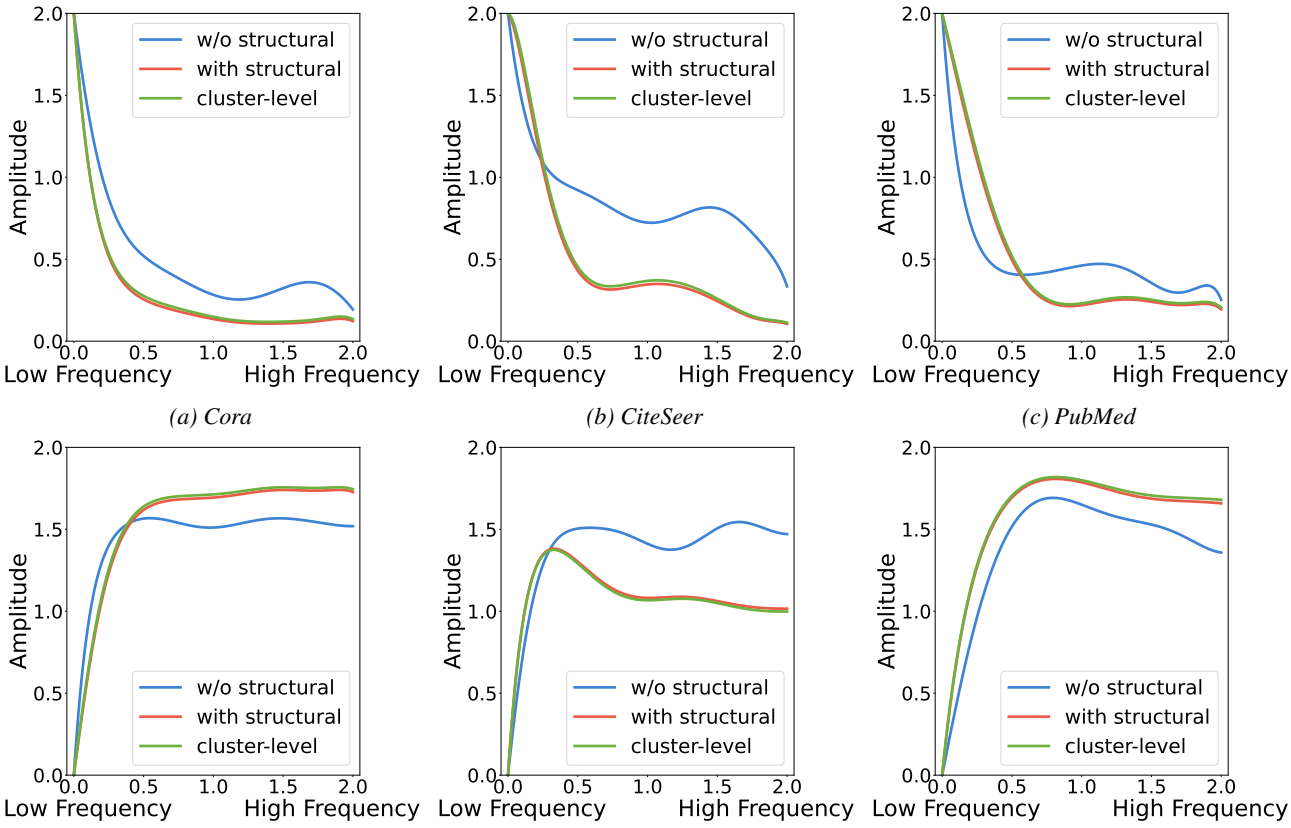


(a) Cora *(b) CiteSeer* *(c) PubMed*

(d) Roman-empire *(e) Tolokers* *(f) Questions*

Figure 12. Semantic representations on six datasets under overlapping partitioning setting with 50 clients.

both client side and server side. A detailed analysis of time complexity is provided in Appendix E.



(a) Cora *(b) CiteSeer* *(c) PubMed*

(d) Roman-empire *(e) Tolokers* *(f) Questions*

Figure 13. Spectral properties on six datasets under overlapping partitioning setting with 50 clients.

Table 8. Ablation studies are conducted under both non-overlapping and overlapping partitioning settings on eleven datasets.

Cora							
Semantic	Structural	non-overlapping 5 clients	non-overlapping 10 clients	non-overlapping 20 clients	overlapping 10 clients	overlapping 30 clients	overlapping 50 clients
✔	✗	80.07±0.50 (↓ 4.60)	78.52±0.69 (↓ 3.80)	80.44±0.11 (↓ 3.69)	76.91±0.89 (↓ 4.11)	72.33±0.39 (↓ 4.56)	75.11±0.13 (↓ 3.47)
✗	✔	79.32±0.32 (↓ 5.35)	78.30±1.11 (↓ 4.02)	79.89±0.09 (↓ 4.24)	75.99±1.00 (↓ 5.03)	72.17±0.51 (↓ 4.72)	74.80±0.21 (↓ 3.78)
✗	✗	78.85±0.15 (↓ 5.82)	78.03±1.04 (↓ 4.29)	79.63±0.56 (↓ 4.50)	75.39±0.55 (↓ 5.63)	71.84±0.74 (↓ 5.05)	74.65±0.19 (↓ 3.93)
✔	✔	**84.67±0.05**	**82.32±0.04**	**84.13±0.09**	**81.02±0.09**	**76.89±0.06**	**78.58±0.15**
CiteSeer							
Semantic	Structural	non-overlapping 5 clients	non-overlapping 10 clients	non-overlapping 20 clients	overlapping 10 clients	overlapping 30 clients	overlapping 50 clients
✔	✗	69.76±1.28 (↓ 3.30)	74.08±0.23 (↓ 3.57)	71.48±0.94 (↓ 2.61)	68.11±0.54 (↓ 4.08)	68.36±0.08 (↓ 4.02)	66.38±1.13 (↓ 3.30)
✗	✔	69.41±0.19 (↓ 3.65)	73.05±0.68 (↓ 4.60)	70.82±1.27 (↓ 3.27)	67.72±1.09 (↓ 4.47)	67.42±0.11 (↓ 4.96)	65.84±0.36 (↓ 3.84)
✗	✗	69.20±0.08 (↓ 3.86)	72.80±0.44 (↓ 4.85)	69.51±1.18 (↓ 4.58)	66.78±0.58 (↓ 5.41)	67.14±0.51 (↓ 5.24)	65.63±0.30 (↓ 4.05)
✔	✔	**73.06±0.08**	**77.65±0.10**	**74.09±0.09**	**72.19±0.08**	**72.38±0.13**	**69.68±0.10**
PubMed							
Semantic	Structural	non-overlapping 5 clients	non-overlapping 10 clients	non-overlapping 20 clients	overlapping 10 clients	overlapping 30 clients	overlapping 50 clients
✔	✗	85.05±0.30 (↓ 3.06)	84.73±0.18 (↓ 3.05)	84.31±0.14 (↓ 3.06)	83.54±0.15 (↓ 2.86)	82.47±0.73 (↓ 4.36)	81.73±0.30 (↓ 3.47)
✗	✔	84.64±0.56 (↓ 3.47)	83.31±0.32 (↓ 4.47)	83.65±0.24 (↓ 3.72)	82.40±0.03 (↓ 4.00)	81.70±0.53 (↓ 5.13)	81.22±0.76 (↓ 3.98)
✗	✗	84.37±0.16 (↓ 3.74)	83.12±0.20 (↓ 4.66)	83.24±0.28 (↓ 4.13)	81.79±0.29 (↓ 4.61)	80.95±0.73 (↓ 5.88)	80.72±0.65 (↓ 4.48)
✔	✔	**88.11±0.07**	**87.78±0.13**	**87.37±0.14**	**86.40±0.09**	**86.83±0.06**	**85.20±0.04**
Amazon-Computer							
Semantic	Structural	non-overlapping 5 clients	non-overlapping 10 clients	non-overlapping 20 clients	overlapping 10 clients	overlapping 30 clients	overlapping 50 clients
✔	✗	88.53±0.10 (↓ 2.56)	87.61±0.22 (↓ 3.69)	88.88±0.84 (↓ 1.74)	87.62±0.27 (↓ 2.79)	87.06±0.11 (↓ 2.59)	86.29±0.33 (↓ 3.05)
✗	✔	88.18±0.74 (↓ 2.91)	86.95±0.79 (↓ 4.35)	88.42±0.73 (↓ 2.20)	87.40±0.40 (↓ 3.01)	86.32±0.20 (↓ 3.33)	85.32±0.71 (↓ 4.02)
✗	✗	87.72±0.13 (↓ 3.37)	86.74±0.53 (↓ 4.56)	88.32±0.31 (↓ 2.30)	87.21±0.23 (↓ 3.20)	86.24±0.55 (↓ 3.41)	85.13±0.19 (↓ 4.21)
✔	✔	**91.09±0.07**	**91.30±0.13**	**90.62±0.09**	**90.41±0.07**	**89.65±0.12**	**89.34±0.11**
Amazon-Photo							
Semantic	Structural	non-overlapping 5 clients	non-overlapping 10 clients	non-overlapping 20 clients	overlapping 10 clients	overlapping 30 clients	overlapping 50 clients
✔	✗	90.65±0.09 (↓ 3.21)	91.38±0.90 (↓ 3.24)	90.60±0.15 (↓ 3.16)	90.21±0.60 (↓ 3.51)	91.61±0.72 (↓ 2.83)	90.78±0.77 (↓ 2.58)
✗	✔	90.37±0.45 (↓ 3.49)	91.01±0.24 (↓ 3.61)	90.24±0.44 (↓ 3.52)	89.84±0.50 (↓ 3.88)	91.21±0.18 (↓ 3.23)	90.31±0.09 (↓ 3.05)
✗	✗	89.63±0.29 (↓ 4.23)	90.89±0.08 (↓ 3.73)	89.85±0.43 (↓ 3.91)	89.62±0.27 (↓ 4.10)	90.35±0.52 (↓ 4.09)	89.89±0.38 (↓ 3.47)
✔	✔	**93.86±0.15**	**94.62±0.14**	**93.76±0.07**	**93.72±0.15**	**94.44±0.11**	**93.36±0.14**
ogbn-arxiv							
Semantic	Structural	non-overlapping 5 clients	non-overlapping 10 clients	non-overlapping 20 clients	overlapping 10 clients	overlapping 30 clients	overlapping 50 clients
✔	✗	67.29±0.50 (↓ 3.57)	66.26±0.67 (↓ 3.21)	65.53±0.06 (↓ 3.24)	63.16±0.14 (↓ 4.28)	63.64±0.86 (↓ 2.74)	62.86±0.34 (↓ 2.87)
✗	✔	67.19±0.17 (↓ 3.67)	66.13±0.50 (↓ 3.34)	65.40±0.49 (↓ 3.37)	62.99±0.38 (↓ 4.45)	63.07±0.83 (↓ 3.31)	62.13±0.36 (↓ 3.60)
✗	✗	66.86±0.59 (↓ 4.00)	65.96±0.12 (↓ 3.51)	64.79±0.08 (↓ 3.98)	62.17±0.57 (↓ 5.27)	62.39±0.55 (↓ 3.99)	61.65±0.08 (↓ 4.08)
✔	✔	**70.86±0.13**	**69.47±0.08**	**68.77±0.13**	**67.44±0.12**	**66.38±0.13**	**65.73±0.07**
Roman-empire							
Semantic	Structural	non-overlapping 5 clients	non-overlapping 10 clients	non-overlapping 20 clients	overlapping 10 clients	overlapping 30 clients	overlapping 50 clients
✔	✗	63.15±0.64 (↓ 5.52)	62.67±0.14 (↓ 4.14)	62.53±0.20 (↓ 2.61)	59.85±0.54 (↓ 5.81)	60.28±0.14 (↓ 3.52)	58.70±0.49 (↓ 4.05)
✗	✔	62.86±0.51 (↓ 5.81)	62.46±0.13 (↓ 4.35)	61.97±0.15 (↓ 3.17)	60.48±0.77 (↓ 5.18)	59.60±0.17 (↓ 4.20)	58.37±0.13 (↓ 4.38)
✗	✗	62.37±0.74 (↓ 6.30)	61.76±0.42 (↓ 5.05)	61.63±0.18 (↓ 3.51)	59.22±0.84 (↓ 6.44)	59.16±0.47 (↓ 4.64)	58.13±0.45 (↓ 4.62)
✔	✔	**68.67±0.10**	**66.81±0.09**	**65.14±0.15**	**65.66±0.07**	**63.80±0.09**	**62.75±0.14**
Amazon-ratings							
Semantic	Structural	non-overlapping 5 clients	non-overlapping 10 clients	non-overlapping 20 clients	overlapping 10 clients	overlapping 30 clients	overlapping 50 clients
✔	✗	41.78±0.12 (↓ 3.40)	41.29±0.20 (↓ 3.82)	42.57±0.16 (↓ 3.56)	39.46±0.17 (↓ 3.37)	39.19±0.28 (↓ 3.33)	40.67±0.40 (↓ 2.30)
✗	✔	40.94±0.33 (↓ 4.24)	41.23±0.37 (↓ 3.88)	42.28±0.27 (↓ 3.85)	39.40±0.24 (↓ 3.43)	38.89±0.25 (↓ 3.63)	40.51±0.15 (↓ 2.46)
✗	✗	40.52±0.22 (↓ 4.66)	41.05±0.21 (↓ 4.06)	42.05±0.13 (↓ 4.08)	38.62±0.13 (↓ 4.21)	38.78±0.19 (↓ 3.74)	40.03±0.70 (↓ 2.94)
✔	✔	**45.18±0.14**	**45.11±0.15**	**46.13±0.05**	**42.83±0.06**	**42.52±0.10**	**42.97±0.15**
Minesweeper							
Semantic	Structural	non-overlapping 5 clients	non-overlapping 10 clients	non-overlapping 20 clients	overlapping 10 clients	overlapping 30 clients	overlapping 50 clients
✔	✗	80.90±0.71 (↓ 1.36)	80.13±0.07 (↓ 2.03)	79.87±0.12 (↓ 2.73)	72.62±0.13 (↓ 3.03)	70.46±0.30 (↓ 2.14)	68.60±0.44 (↓ 2.71)
✗	✔	80.72±0.33 (↓ 1.54)	79.49±0.39 (↓ 2.67)	79.78±0.37 (↓ 2.82)	72.15±0.52 (↓ 3.50)	70.09±0.38 (↓ 2.51)	68.64±0.10 (↓ 2.67)
✗	✗	79.56±0.18 (↓ 2.70)	78.84±0.44 (↓ 3.32)	79.27±0.46 (↓ 3.33)	71.26±0.38 (↓ 4.39)	69.37±0.26 (↓ 3.23)	67.31±0.39 (↓ 4.00)
✔	✔	**82.26±0.14**	**82.16±0.08**	**82.60±0.11**	**75.65±0.11**	**72.60±0.06**	**71.31±0.05**
Tolokers							
Semantic	Structural	non-overlapping 5 clients	non-overlapping 10 clients	non-overlapping 20 clients	overlapping 10 clients	overlapping 30 clients	overlapping 50 clients
✔	✗	71.88±0.42 (↓ 3.94)	70.33±0.58 (↓ 3.63)	66.27±0.28 (↓ 4.61)	71.71±0.31 (↓ 2.62)	68.36±0.42 (↓ 3.91)	68.96±0.40 (↓ 2.07)
✗	✔	71.37±0.14 (↓ 4.45)	70.08±0.46 (↓ 3.88)	66.14±0.42 (↓ 4.74)	71.47±0.22 (↓ 2.86)	68.16±0.64 (↓ 4.11)	68.45±0.31 (↓ 2.58)
✗	✗	70.11±0.26 (↓ 5.71)	69.61±0.32 (↓ 4.35)	65.59±0.37 (↓ 5.29)	70.59±0.19 (↓ 3.74)	67.09±0.46 (↓ 5.18)	68.14±0.61 (↓ 2.89)
✔	✔	**75.82±0.05**	**73.96±0.10**	**70.88±0.12**	**74.33±0.09**	**72.27±0.07**	**71.03±0.09**
Questions							
Semantic	Structural	non-overlapping 5 clients	non-overlapping 10 clients	non-overlapping 20 clients	overlapping 10 clients	overlapping 30 clients	overlapping 50 clients
✔	✗	66.94±0.23 (↓ 2.57)	65.44±0.63 (↓ 3.25)	62.33±0.59 (↓ 3.41)	67.82±0.47 (↓ 1.57)	64.55±0.12 (↓ 1.88)	61.63±0.19 (↓ 3.31)
✗	✔	66.72±0.45 (↓ 2.79)	65.37±0.26 (↓ 3.32)	61.51±0.32 (↓ 4.23)	67.53±0.26 (↓ 1.86)	64.29±0.35 (↓ 2.14)	61.48±0.17 (↓ 3.46)
✗	✗	65.70±0.18 (↓ 3.81)	64.89±0.44 (↓ 3.80)	61.23±0.22 (↓ 4.51)	66.33±0.53 (↓ 3.06)	63.62±0.31 (↓ 2.81)	60.49±0.37 (↓ 4.45)
✔	✔	**69.51±0.15**	**68.69±0.18**	**65.74±0.07**	**69.39±0.05**	**66.43±0.13**	**64.94±0.12**

Table 9. Comparison between single Gaussian approximation and GMM-based approximation on *Cora* and *Roman-empire* datasets. The cross (*i.e.*, $\times$) denotes that GMM-based approximation is not significantly better than single Gaussian approximation revealed by a paired t-test with significance level 0.05.

Cora						
Methods	non-overlapping 5 clients	non-overlapping 10 clients	non-overlapping 20 clients	overlapping 10 clients	overlapping 30 clients	overlapping 50 clients
single Gaussian approximation	84.67±0.05	82.32±0.04	84.13±0.09	81.02±0.09	76.89±0.06	78.58±0.15
GMM-based approximation	84.65±0.14 ×	82.44±0.15 ×	84.32±0.13 ×	81.20±0.16 ×	76.97±0.12 ×	78.69±0.17 ×
Roman-empire						
Methods	non-overlapping 5 clients	non-overlapping 10 clients	non-overlapping 20 clients	overlapping 10 clients	overlapping 30 clients	overlapping 50 clients
single Gaussian approximation	68.67±0.10	66.81±0.09	65.14±0.15	65.66±0.07	63.80±0.09	62.75±0.14
GMM-based approximation	68.73±0.15 ×	67.04±0.17 ×	65.29±0.19 ×	65.78±0.20 ×	63.96±0.18 ×	62.91±0.21 ×

Table 10. Time consumption (seconds) of each communication round for our proposed FedSSA and baseline methods on *Cora* and *Roman-empire* datasets.

	Cora					
Methods	non-overlapping 5 clients	non-overlapping 10 clients	non-overlapping 20 clients	overlapping 10 clients	overlapping 30 clients	overlapping 50 clients
FedAvg (McMahan et al., 2017)	5.51	2.19	5.63	5.40	7.27	12.08
FedProx (Li et al., 2020)	4.24	4.65	8.86	5.49	13.71	22.43
FedPer (Arivazhagan et al., 2019)	4.06	4.13	8.17	4.16	12.38	20.24
GCFL (Xie et al., 2021)	6.30	9.38	18.87	9.78	27.93	46.22
FedGNN (Wu et al., 2021)	2.28	4.42	8.76	5.40	13.07	23.04
FedSage+(Zhang et al., 2021)	6.88	8.55	17.88	9.37	16.97	23.35
FED-PUB (Baek et al., 2023)	22.04	27.34	60.31	33.46	80.54	147.84
FedGTA (Li et al., 2023)	3.36	2.24	5.89	4.30	5.00	7.18
AdaFGL (Li et al., 2024)	1.99	2.49	4.63	4.44	6.16	7.83
FedTAD (Zhu et al., 2024)	4.91	5.25	9.13	5.22	12.84	19.71
FedIIH (Yu et al., 2025a)	19.57	22.76	56.09	19.67	65.49	139.03
FedSSA (Ours)	2.87	3.46	4.94	6.48	11.17	16.47
Average	7.00	8.07	17.43	9.43	22.71	40.45
	Roman-empire					
Methods	non-overlapping 5 clients	non-overlapping 10 clients	non-overlapping 20 clients	overlapping 10 clients	overlapping 30 clients	overlapping 50 clients
FedAvg (McMahan et al., 2017)	8.20	5.76	8.46	10.25	18.47	19.12
FedProx (Li et al., 2020)	4.31	6.73	9.04	6.90	16.25	23.66
FedPer (Arivazhagan et al., 2019)	5.35	6.19	9.19	6.47	15.58	20.22
GCFL (Xie et al., 2021)	6.32	9.58	18.37	10.84	29.91	50.23
FedGNN (Wu et al., 2021)	3.33	6.60	9.43	6.45	15.29	22.82
FedSage+(Zhang et al., 2021)	10.06	14.82	26.27	23.09	47.42	62.72
FED-PUB (Baek et al., 2023)	18.81	28.03	61.75	28.45	83.07	133.05
FedGTA (Li et al., 2023)	2.17	3.58	5.32	3.42	6.12	9.92
AdaFGL (Li et al., 2024)	4.34	4.14	5.55	6.49	7.80	10.69
FedTAD (Zhu et al., 2024)	4.88	8.55	15.03	10.98	22.42	37.01
FedIIH (Yu et al., 2025a)	17.45	24.90	47.01	28.19	61.17	100.10
FedSSA (Ours)	5.78	6.01	11.93	9.04	16.59	24.78
Average	7.58	10.41	18.95	12.55	28.34	42.86