FedMed: Federated Learning-Based Personalized and Safe Medication Recommendation

Anchen Li*
Department of Computer Science,
Aalto University
Espoo, Finland
anchen.li@aalto.fi

Elena Casiraghi
Department of Computer Science,
Aalto University
Espoo, Finland
Department of Computer Science,
University of Milan
Milan, Italy
Environmental Genomics and
Systems Biology Division, Lawrence
Berkeley National Lab
Berkeley, United States
elena.casiraghi@unimi.it

Juho Rousu
Department of Computer Science,
Aalto University
Espoo, Finland
Helsinki Institute for Information
Technology, University of Helsinki
Helsinki, Finland
juho.rousu@aalto.fi

Abstract

Medication recommendation for patients with multimorbidity is an essential task for AI in healthcare. Existing works rely on centralized storage of all patient clinical records (e.g., diagnoses and procedures) to develop a centrally managed medication recommendation model. However, this setting exposes sensitive patient data to privacy risks and potential leakage. To address this issue, we propose FedMed, a privacy-preserving, personalized, and safe medication recommender based on federated learning. In FedMed, each patient client develops a prediction module using its private clinical records for medication recommendation, and a self-supervised strategy is further introduced to alleviate record data sparsity. On the server side, the global server performs a personalized graph-guided aggregation mechanism for updating and synchronizing client public parameters without accessing patient private information, thereby preserving privacy. In addition, our FedMed mitigates adverse interactions between medications, known as drug-drug interaction (DDI), by incorporating DDI regularization at the client and server sides to promote predicted medication safety. The results on the two Electronic Health Records (EHR) datasets show the effectiveness of our federated medication recommender FedMed.

CCS Concepts

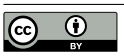
• **Information systems** → **Recommender systems**; • **Applied computing** → **Health care information systems**.

Keywords

Drug Recommendation; Health Informatics; Federated Learning.

ACM Reference Format:
Anchen Li, Elena Casiraghi, and Juho Rousu. 2026. FedMed: Federated Learning-Based Personalized and Safe Medication Recommendation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '26), August 09–13, 2026, Jeju Island, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3770855.3818934

*Corresponding author.



KDD '26, Jeju Island, Republic of Korea

ACM ISBN 979-8-4007-2259-2/2026/08
https://doi.org/10.1145/3770855.3818934

1 Introduction

Multimorbidity, where a patient has multiple chronic or acute medical conditions, is increasingly common and creates major challenges for healthcare, particularly in medication management [31, 40]. For such patients, the medication recommendation task aims to select an appropriate combination of drugs based on their clinical records (e.g., diagnoses and procedures) from hospital visits [37]. Meanwhile, the growing availability of healthcare data (e.g., electronic health records) provides new opportunities to develop more accurate medication recommender systems. This trend has also driven the emergence of commercial health IT platforms (e.g., Merative[1]).

To recommend proper drugs, traditional approaches are mainly based on rule-driven protocols developed by clinicians following medical guidelines or their own experience [3, 7], but these manually designed protocols are time-consuming to develop and difficult to update or apply in practice. Recently, researchers propose deep learning-based medication recommenders that use different neural components (e.g., transformers [48], recurrent neural networks [43], and memory networks [34]) to model patient clinical records and achieve better performance than traditional methods.

Notably, most existing medication recommenders rely on centralized storage of all patient clinical records for model training. Since patient data contains sensitive information, this centralized setting may lead to privacy risks and has raised increasing social and regulatory concerns (e.g., General Data Protection Regulation [32]). To this end, we adopt federated learning [22], a privacy-preserving paradigm that avoids sharing private patient data, for medication recommendation. In this paradigm, each local patient client trains the recommender model using its private clinical records, while a global server coordinates the training process by synchronizing public parameters across clients. As a result, privacy protection is achieved because patient information remains on local clients.

However, deploying medication recommendation in the federated learning framework faces three challenges. First, since clients

[1]https://www.merative.com/

in federated learning settings usually have limited computational resources [12, 23], client models should be simple and lightweight. Thus, directly adapting existing medication recommender models [34, 42, 43, 48, 50], which often use complex neural architectures, can be computationally demanding and may not fit well in such settings. Moreover, each client is trained only on its local patient records, which are often sparse and not enough for effective client training, especially for complex models that can easily overfit and show worse performance. Second, most federated learning methods use averaging aggregation ideas [24, 29] on the server for client updates, which overlook patient latent correlations and fail to support personalized refinement of patient clients. Third, the recommended medication set should ensure safety by mitigating harmful drug-drug interactions (DDI), which requires careful integration into the design of both patient clients and the global server.

To this end, we propose FedMed, a privacy-preserving, personalized, and safe federated medication recommender. On the client side, we consider patient diagnoses and procedures as private data and model them using a simple yet effective attention-based multi-pooling module to represent patient health conditions, which then interact with a public drug embedding parameter matrix to generate medication recommendation predictions, and we further design a self-supervised learning strategy based on diagnosis and procedure records to provide additional supervision signals. On the server side, we use public drug parameters from clients to build a patient relation graph without privacy exposure and design a graph-guided aggregation mechanism for client synchronization and updating, which promotes personalized patient modeling. We also introduce DDI regularization on both client and server sides to mitigate harmful adverse interactions in recommended medication sets. To evaluate our method FedMed, we perform experiments on two real-world EHR datasets, and results show that our FedMed achieves strong performance. Ablation studies further validate the rationality of the components designed in FedMed.

To summarize, our contributions in this work are as follows:

- We propose FedMed, a privacy-preserving, personalized, and safe medication recommender, which introduces and investigates federated learning in medication recommendation.
- We address three challenges in federated medication recommendation by introducing self-supervised learning on the client to alleviate data sparsity, graph-guided aggregation on the server for personalized client updating, and DDI regularization on client and server sides to improve medication recommendation safety.
- Extensive experimental results on two real-world EHR datasets demonstrate the effectiveness of our FedMed framework.

2 Preliminaries

In this section, we define some definitions and introduce the background for medication recommendation and federated learning.

2.1 Medication Recommendation

Given patient set $\mathcal{U}$, diagnosis set $\mathcal{D}$, procedure set $\mathcal{P}$, and medication set $\mathcal{M}$ with their elements $u \in \mathcal{U}$, $d \in \mathcal{D}$, $p \in \mathcal{P}$, and $m \in \mathcal{M}$, we introduce the data and problem formulation for this task:

Electronic Health Records (EHR): such data is denoted as a set $\mathcal{X}$, where element $\mathcal{X}_u \in \mathcal{X}$ is the EHR of patient $u \in \mathcal{U}$. For $\mathcal{X}_u$, it is also a set, and each element $x_u^i \in \mathcal{X}_u$ denotes a visit of patient u. For visit x_u^i, it is defined as $x_u^i = \{\mathcal{D}_u^i, \mathcal{P}_u^i, \mathcal{M}_u^i\}$, where $\mathcal{D}_u^i \subset \mathcal{D}$, $\mathcal{P}_u^i \subset \mathcal{P}$, and $\mathcal{M}_u^i \subset \mathcal{M}$ are the diagnosis, procedure, and medication records associated with visit x_u^i, respectively.

Drug-Drug Interaction (DDI) Matrix: a symmetric matrix $\mathbf{T} \in \{0,1\}^{|\mathcal{M}|\times|\mathcal{M}|}$ is utilized to denote adverse DDI, where $t_{ij} \in \mathbf{T}$ equals 1 if medication m_i and m_j interact, and $t_{ij} = 0$ otherwise.

Problem Formulation: for patient u, given entire historical EHR $\mathcal{X}$ and DDI matrix $\mathbf{T}$, along with diagnosis and procedure sets (i.e., $\mathcal{D}_u$ and $\mathcal{P}_u$) at the target visit of patient u, general medication recommendation aims to predict medication set $\mathcal{M}_u$ for this visit.

2.2 Federated Learning

As a secure distributed paradigm, federated learning trains a global server model to support multiple client models while keeping their data private. To be specific, given N clients, the goal of federated learning [23, 29] is to minimize the accumulated loss of all clients as: $\min \mathcal{L} = \sum_{i=1}^{N} \alpha_i \mathcal{L}_i(\theta)$, where $\mathcal{L}_i(\cdot)$ and α_i are the loss function and weight for the i-th client, and θ is the client parameter. We design our method based on this federated learning framework.

3 Methodology

We propose FedMed, a medication recommender based on federated learning, which contains two key components, the client and server. As shown in Figure 1, each communication round of FedMed has four steps. First, client model training is performed. Second, clients upload relevant parameters to the server. Third, the server aggregates and updates the received parameters from the clients. Fourth, the server distributes the updated parameters back to the clients. In what follows, we first introduce the four steps of our FedMed, then describe the overall optimization and complexity analysis.

3.1 Client Model Training (Step 1)

We treat each patient as a client under the federated learning setting. To be specific, each patient client model includes three key parts: a medication prediction module, a contrastive learning strategy, and a drug-drug interaction (DDI) constraint.

3.1.1 ***Medication Prediction Module.*** For patient u, this module learns a function $\mathcal{F}_u$ that takes diagnosis and procedure records ($\mathcal{D}_u$ and $\mathcal{P}_u$) as input to predict the required medication set $\hat{\mathcal{M}}_u$:

$$\hat{\mathcal{M}}_u = \mathcal{F}_u(\mathcal{D}_u, \mathcal{P}_u | \mathcal{X}_u, \mathbf{T}, \Theta_u), \tag{1}$$

where Θ_u is the parameters of $\mathcal{F}_u$ and $\mathcal{X}_u$ is the EHR of patient u.

We can observe that, compared with the general medication recommendation models (i.e., the problem formulation in Section 2.1), our module (i.e., Eq.(1)) utilizes patient's own records (i.e., $\mathcal{X}_u$) and the local model parameter (i.e., Θ_u) for prediction, which avoids the exposure of sensitive data and protects patient privacy.

Some existing medication recommendation models [42, 43, 48, 50] can be directly applied to this module. However, many of them adopt complex architectures. In federated learning, the client model should be lightweight, as client devices typically have limited computational resources. Moreover, patient records on each client are often sparse, which makes complex models prone to overfitting. Therefore, we design a simple and effective attention-based multi-pooling (AMP) module to implement the function $\mathcal{F}_u$. Specifically,

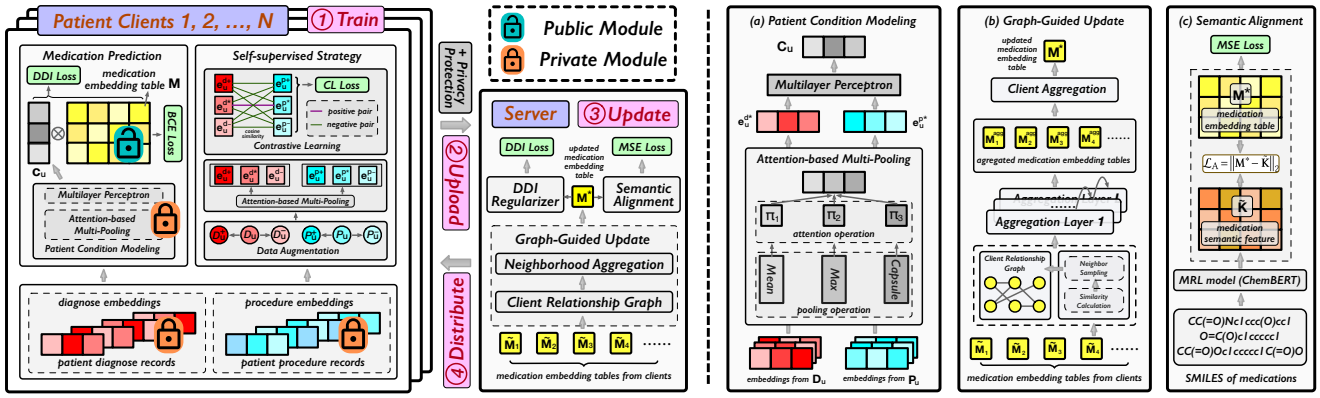


Figure 1: Left: The overall FedMed framework, with four steps per communication round. Right: Details of three components.

for patient u, AMP models patient's diagnosis and procedure representations $\mathbf{e}_u^{d*}$ and $\mathbf{e}_u^{p*}$ based on $\mathcal{D}_u$ and $\mathcal{P}_u$, as follows:

$$\begin{aligned}\mathbf{e}_u^{d*} &= \mathrm{AMP}(\mathcal{D}_u) = \sum\nolimits_{\mathrm{Pool}_\psi \in \Psi} \pi_\psi^{\mathcal{D}} \cdot \mathrm{Pool}_\psi\left(\mathbf{d}_i, \forall d_i \in \mathcal{D}_u\right),\\ \mathbf{e}_u^{p*} &= \mathrm{AMP}(\mathcal{P}_u) = \sum\nolimits_{\mathrm{Pool}_\psi \in \Psi} \pi_\psi^{\mathcal{P}} \cdot \mathrm{Pool}_\psi\left(\mathbf{p}_i, \forall p_i \in \mathcal{P}_u\right),\end{aligned} \tag{2}$$

where $\mathbf{d}_i$ and $\mathbf{p}_i$ are embeddings looked up by IDs of diagnosis d_i and procedure p_i; $\mathrm{Pool}_\psi \in \Psi$ is a pooling function, and function set Ψ contains three operations (i.e., mean, max, capsule poolings), which model the set structured data from different perspectives. Mean and max poolings are commonly used, whereas capsule pooling follows the formulation in [33]; $\pi^{\mathcal{D}}$ and $\pi^{\mathcal{P}}$ are attention weights, and $\pi^{\mathcal{D}}$ is defined as follows:

$$\pi_\psi^{\mathcal{D}} = \frac{\exp\left(\mathbf{w}^\top \mathrm{Pool}_\psi\left(\mathbf{d}_i, \forall d_i \in \mathcal{D}_u\right)\right)}{\sum_{\mathrm{Pool}_{\psi'} \in \Psi} \exp\left(\mathbf{w}^\top \mathrm{Pool}_{\psi'}\left(\mathbf{d}_i, \forall d_i \in \mathcal{D}_u\right)\right)}, \tag{3}$$

where $\mathbf{w}$ is the weight matrix in the attention mechanism. The definition of $\pi^{\mathcal{P}}$ is similar to $\pi^{\mathcal{D}}$ and is omitted for brevity.

Then, we use an MLP (e.g., two layers) to fuse the patient diagnosis and procedure representations $\mathbf{e}_u^{d*}$ and $\mathbf{e}_u^{p*}$ to obtain the patient state representation $\mathbf{c}_u$, which is then interacted with the medication embedding table $\mathbf{M}$ and passed through a sigmoid function σ to produce the predicted medication representation $\hat{\mathbf{m}}_u$, as:

$$\hat{\mathbf{m}}_u = \sigma(\mathbf{M}^\top \mathbf{c}_u) \in \mathbb{R}^{|\mathcal{M}|}, \quad \mathbf{c}_u = \mathrm{MLP}(\mathbf{e}_u^{d*} \| \mathbf{e}_u^{p*}), \tag{4}$$

where $\|$ denotes the vector concatenation operation. Here, $\mathbf{M}$ is updated on the server side and then distributed to all clients in the previous epoch, which will be described in Sections 3.3 and 3.4.

During training, following [43], we first convert ground-truth medication set $\mathcal{M}_u$ into a multi-hot vector $\mathbf{m}_u \in \{0,1\}^{|\mathcal{M}|}$, where each entry indicates whether the corresponding medication appears in $\mathcal{M}_u$. We then use $\mathbf{m}_u$ to penalize predicted representation $\hat{\mathbf{m}}_u$:

$$\mathcal{L}_{\mathrm{REC}} = -\sum\nolimits_{i=1}^{|\mathcal{M}|} \left[m_{u,i} \cdot \log\left(\hat{m}_{u,i}\right) + \left(1 - m_{u,i}\right) \cdot \log\left(1 - \hat{m}_{u,i}\right)\right], \tag{5}$$

where $m_{u,i}$ and $\hat{m}_{u,i}$ are the i-th elements of vectors $\mathbf{m}_u$ and $\hat{\mathbf{m}}_u$.

3.1.2 ***Self-supervised Strategy.*** As discussed in Introduction section, client training may suffer from the data sparsity issue. To this end, we use a self-supervised strategy [13, 17, 19, 20, 27] that includes data augmentation and contrastive learning to provide auxiliary supervision signals for model training.

First, for data augmentation, we create different views of patient u diagnosis and procedure records (i.e., sets $\mathcal{D}_u$ and $\mathcal{P}_u$) using two operations, Drop and Add, which randomly drop and add elements:

$$\mathcal{D}_u^- = \mathrm{Drop}(\mathcal{D}_u),\ \mathcal{D}_u^+ = \mathrm{Add}(\mathcal{D}_u),\ \mathcal{P}_u^- = \mathrm{Drop}(\mathcal{P}_u),\ \mathcal{P}_u^+ = \mathrm{Add}(\mathcal{P}_u). \tag{6}$$

Based on our AMP operation (i.e., Eq.(2)), we obtain the corresponding representations $\mathbf{e}_u^{d+}$, $\mathbf{e}_u^{d-}$, $\mathbf{e}_u^{p+}$, and $\mathbf{e}_u^{p-}$, as follows:

$$\mathbf{e}_u^{d+} = \mathrm{AMP}(\mathcal{D}_u^+),\ \mathbf{e}_u^{d-} = \mathrm{AMP}(\mathcal{D}_u^-),\ \mathbf{e}_u^{p+} = \mathrm{AMP}(\mathcal{P}_u^+),\ \mathbf{e}_u^{p-} = \mathrm{AMP}(\mathcal{P}_u^-) \tag{7}$$

which serve as augmented representations of $\mathbf{e}_u^{d*}$ and $\mathbf{e}_u^{p*}$ in Eq.(2).

Then, based on four vectors above and original representations $\mathbf{e}_u^{d*}$ and $\mathbf{e}_u^{p*}$, a contrastive loss is designed. We treat $(\mathbf{e}_u^{d*}, \mathbf{e}_u^{p*})$ as a positive pair, and all other pairs as negative. The supervision from positive pairs encourages consistent and aligned representations of diagnosis and procedure records, while negative pairs improve the discriminative ability of the model. Following [5], we employ the InfoNCE loss [8] to increase the agreement between positive pairs and reduce that between negative pairs, as follows:

$$\mathcal{L}_{\mathrm{SSL}} = -\log \frac{\exp(c(\mathbf{e}_u^{d*}, \mathbf{e}_u^{p*}))}{\sum_{\star \in \{*,+,-\}} \exp(c(\mathbf{e}_u^{d\star}, \mathbf{e}_u^{p\star}))}, \tag{8}$$

where c denotes the cosine similarity between two embeddings.

3.1.3 ***DDI Regularization.*** To mitigate the adverse interactions among the recommended medications, we use a DDI constraint term [34, 43] in the client model. Given predicted medication vector representation $\hat{\mathbf{m}}_u$ (in Eq.(4)), the client DDI loss is defined as:

$$\mathcal{L}_{\mathrm{DDI}}^c = \sum\nolimits_{i=1}^{|\mathcal{M}|} \sum\nolimits_{j=i+1}^{|\mathcal{M}|} \mathbb{I}(t_{ij} = 1) \cdot \hat{m}_{u,i} \cdot \hat{m}_{u,j}, \tag{9}$$

where t_{ij} is the element of the DDI matrix $\mathbf{T}$ (introduced in Section 2.1), $\mathbb{I}$ denotes the indicator function, and $\hat{m}_{u,i}$ and $\hat{m}_{u,j}$ are the i-th and j-th elements of the vector $\hat{\mathbf{m}}_u$, respectively.

3.1.4 ***Loss Function on Client.*** For patient u, the loss of its client model contains the medication recommendation loss (i.e., Eq.(5)), self-supervised loss (i.e., Eq.(8)), and DDI loss (i.e., Eq.(9)), as:

$$\mathcal{L}_{\mathrm{client}(u)}(\Theta_u) = \mathcal{L}_{\mathrm{REC}} + \lambda_{\mathrm{SSL}} \mathcal{L}_{\mathrm{SSL}} + \lambda_{\mathrm{DDI}}^c \mathcal{L}_{\mathrm{DDI}}^c, \tag{10}$$

where λ_{SSL} and λ_{DDI}^c adjust the contrastive learning and DDI constraint strength, respectively, and Θ_u is the client model parameter, including the embeddings of diagnosis, procedure, and medication, as well as the weights in the AMP operation (in Eq.(2)).

3.2 Client Parameter Upload (Step 2)

After client training, relevant parameters are uploaded to the server for updates. For medication recommendation, to enhance patient privacy, our client only uploads the public medication embedding table (i.e., $\mathbf{M}$), while patient diagnosis and procedure embeddings and patient records (i.e., X_u) are kept locally on the client.

Although uploading only the medication embedding table greatly improves privacy, the server may still infer private patient information from this table, which could be misused to reveal sensitive data. To this end, inspired by [6], we integrate a differential privacy technique into the client parameter upload step. We perturb medication embedding table $\mathbf{M}$ by injecting element wise Laplacian noise and upload perturbed embedding $\tilde{\mathbf{M}}$ to the server, as:

$$\tilde{\mathbf{M}} = \mathbf{M} + \mathbf{Z}_u, \quad z_{ij} \in \mathbf{Z}_u, \quad z_{ij} \sim \text{Laplace}(0, \delta), \tag{11}$$

where matrix $\mathbf{Z}_u$ has the same shape as $\mathbf{M}$, $z_{ij} \in \mathbf{Z}_u$ is independently sampled from a zero-mean Laplace distribution with scale δ.

Thus, recovering updated medications from changes in medication embeddings is difficult, and privacy improves as δ increases.

3.3 Server Model Updating (Step 3)

The server receives medication embedding table (e.g., $\tilde{\mathbf{M}}$ in Eq.(11)) from all clients. We denote them as a set: $\text{Set}_{\text{med}} = \{\tilde{\mathbf{M}}_1, \cdots, \tilde{\mathbf{M}}_{|\mathcal{U}|}\}$. The server aggregates and updates these parameters via the graph-guided aggregation, semantic alignment, and DDI regularization.

3.3.1 ***Graph-guided Aggregation.*** Traditional server-side federated learning strategies typically rely on FedAvg [29], which treat each client equally and learn unified parameters via average aggregation. However, patients usually share similar characteristics in specific groups, and simple averaging across all patients may hinder personalized modeling for each patient.

To capture patient correlations and enable personalized modeling, we first construct a patient relationship graph $\mathcal{G}$ on the server based on the uploaded parameter set Set_{med} (defined in Section 3.3). The key insight is that patients with similar clinical conditions tend to exhibit similar medication patterns. Moreover, these parameters can be safely shared without revealing private patient information.

For patient i, we define its neighbor set $\mathcal{N}_i$ in $\mathcal{G}$ via the process: we use elements in Set_{med} to compute the cosine similarity between patient i and other patients and obtain a similarity vector $\mathbf{S}_i \in \mathbb{R}^{|\mathcal{U}|}$, and based on $\mathbf{S}_i$, we introduce a sampling strategy to construct the neighborhood $\mathcal{N}_i$ of patient i. Formally, this process is as follows:

$$s_{ij} = c(\tilde{\mathbf{M}}_i, \tilde{\mathbf{M}}_j), \quad s_{ij} \in \mathbf{S}_i, \quad \mathcal{N}_i = \text{TopK}(\mathbf{S}_i) \cup \text{SampleK}\left(\pi(\mathbf{S}_i)\right), \tag{12}$$

where $c(\cdot,\cdot)$ is the cosine similarity; s_{ij} is the similarity between patients i and j; $\text{TopK}(\mathbf{S}_i)$ selects the indices of the top-K largest similarity scores in $\mathbf{S}_i$ as part of neighborhood $\mathcal{N}_i$; $\text{SampleK}(\pi(\mathbf{S}_i))$ samples K additional indices according to probability distribution $\pi(\mathbf{S}_i)$ and adds them to $\mathcal{N}_i$, and $\pi(\cdot)$ is the softmax function, which converts $\mathbf{S}_i$ into a sampling distribution. In this way, our strategy preserves the most similar neighbors while maintaining diversity and exploration, which helps build an informative neighborhood.

Then, based on the constructed neighborhoods of patients, we build a patient relation graph $\mathcal{G}$. Using $\mathcal{G}$, we design a graph-guided aggregation mechanism to update medication embeddings, enabling each client to obtain patient-specific medication representations with the help of neighbors who share similar clinical conditions. Specifically, for patient i, its medication representation $\tilde{\mathbf{M}}_i$ is updated on $\mathcal{G}$ through the neighborhood aggregation [9, 18, 21], as:

$$\tilde{\mathbf{M}}_i^l = \tilde{\mathbf{M}}_i^{l-1} + \frac{1}{|\mathcal{N}_i|} \sum\nolimits_{x \in \mathcal{N}_i} \tilde{\mathbf{M}}_x^{l-1}, \quad \mathbf{M}_i^{\text{agg}} = \frac{1}{L} \sum\nolimits_{l=1}^{L} \tilde{\mathbf{M}}_i^l, \tag{13}$$

where $\tilde{\mathbf{M}}_i^0 = \tilde{\mathbf{M}}_i$ is the initial representation, and l and L represent the aggregation layer index and the total layer number, respectively.

Finally, the server aggregates all neighbor-enhanced medication representations to obtain the unified medication parameter $\mathbf{M}^*$, as:

$$\mathbf{M}^* = \frac{1}{|\mathcal{U}|} \sum\nolimits_{i=1}^{|\mathcal{U}|} \mathbf{M}_i^{\text{agg}}. \tag{14}$$

Compared with FedAvg [29], the above aggregation strategy incorporates information from clinically similar patients and generates more informative medication representations.

3.3.2 ***Semantic Alignment.*** This semantic alignment design optimizes the aggregated medication parameter $\mathbf{M}^*$ (in Eq.(14)) with structural semantic information at the molecular level. We first introduce the SMILES strings [4, 15, 16, 41] of medication molecules (e.g., smiles_i for medication $m_i \in \mathcal{M}$), which encodes molecular chemical structure as a sequence of atoms and bonds, then feed these strings into a molecular representation learning (MRL) model to obtain structural semantic features (i.e., $\mathbf{K}$), and finally design a semantic alignment loss $\mathcal{L}_{\text{A}}$ to incorporate these features into the medication parameter $\mathbf{M}^*$, as follows:

$$\mathbf{K} = \Big\|_{i=1}^{|\mathcal{M}|} \text{MRL}(\text{smiles}_i), \quad \tilde{\mathbf{K}} = \text{TSVD}(\mathbf{K}), \quad \mathcal{L}_{\text{A}} = \left\|\mathbf{M}^* - \tilde{\mathbf{K}}\right\|_2, \tag{15}$$

where MRL is a pretrained model, for which we use ChemBERT [1], that takes the SMILES of a medication as input and outputs its structural semantic representation. The notation $\|_{i=1}^{|\mathcal{M}|}$ denotes concatenation over all medication representations to form the semantic matrix $\mathbf{K}$. Since the dimensionality of $\mathbf{K}$ is inconsistent with that of the medication parameters $\mathbf{M}^*$, we apply truncated singular value decomposition (TSVD) to project $\mathbf{K}$ into the same latent space. Finally, the semantic alignment loss is defined by minimizing the Euclidean distance between matrices $\mathbf{M}^*$ and $\tilde{\mathbf{K}}$.

3.3.3 ***DDI Regularization.*** On the server, we also design a DDI loss to improve the safety of medication recommendations. Specifically, we encourage medications with interactions to be farther apart in the embedding space, and we use this idea to regularize the aggregated medication parameters $\mathbf{M}^*$ (in Eq.(14)), as follows:

$$\mathcal{L}_{\text{DDI}}^s = -\sum\nolimits_{i=1}^{|\mathcal{M}|} \sum\nolimits_{j=i+1}^{|\mathcal{M}|} \mathbb{I}(t_{ij} = 1) \cdot \left\|\mathbf{m}_i^* - \mathbf{m}_j^*\right\|_2, \tag{16}$$

where t_{ij} is the element of the DDI matrix $\mathbf{T}$ (introduced in Section 2.1), $\mathbb{I}$ denotes the indicator function, and vectors $\mathbf{m}_i^*$ and $\mathbf{m}_j^*$ are the i-th and j-th rows of the matrix $\mathbf{M}^*$, which correspond to the representations of medications $m_i \in \mathcal{M}$ and $m_j \in \mathcal{M}$, respectively.

3.3.4 ***Loss Function on Server.*** The server includes the semantic alignment loss (i.e., Eq.(15)) and DDI loss (i.e., Eq.(16)), as follows:

$$\mathcal{L}_{\text{server}}(\Theta_S) = \mathcal{L}_{\text{A}} + \mathcal{L}_{\text{DDI}}^s, \tag{17}$$

where Θ_S denotes the parameters of the server. Since we use a pretrained MRL model that does not involve any learnable parameters, Θ_S mainly consists of the medication parameter $\mathbf{M}^*$ (in Eq.(14)).

3.4 Server Parameter Distribution (Step 4)

After updating the medication parameters $\mathbf{M}^*$ (in Eqs.(12)-(17)), the server distributes them to all clients as the medication embedding table, which is used in the next round of optimization to produce the predicted medication representation in Eq.(4).

3.5 Overall Framework Optimization

Our method FedMed consists of the server and client. For the server, we introduce a loss term $\mathcal{L}_{\text{server}}(\Theta_S)$ based on Eq.(17). For the client, based on Eq.(10), we obtain the overall loss as follows:

$$\min \mathcal{L}_{\text{client}} = \sum\nolimits_{u=1}^{|\mathcal{U}|} \alpha_u \cdot \mathcal{L}_{\text{client}(u)}(\Theta_u), \tag{18}$$

in which $\alpha_u = |\mathcal{X}_u|/|\mathcal{X}|$ denotes the fraction of the patient client training data size relative to the total dataset.

Compared with standard federated learning (introduced in Section 2.2), our client model parameters (i.e., Θ_u) contain both shared parameters (i.e., $\mathbf{M}$), which capture relations across clients, and private parameters (e.g., weights in AMP), which enable personalized and privacy-preserving patient modeling, as detailed in Sections 3.1.4 and 3.2. In addition, on the server side, we introduce a semantic alignment loss and a DDI loss to guide parameter optimization, making the model better suited for medication recommendation. Finally, our FedMed jointly optimizes the client objective $\mathcal{L}_{\text{client}}$ and server objective $\mathcal{L}_{\text{server}}$ in an alternating manner.

3.6 Complexity Analysis

We introduce the model size and time complexity of our method.

3.6.1 ***Time Complexity.*** The time cost of our FedMed mainly comes from the client and server sides. On the client side, the cost is dominated by the medication prediction model, self-supervised strategy, and DDI loss. The time cost is approximately $O\big(|\mathcal{U}|(N_d + N_p)\cdot d + d^2\big)$, where $\mathcal{U}$ is the patient set, N_d and N_p are the average numbers of diagnoses and procedures per patient visit, and d is the embedding size. On the server side, the time cost is from the graph-guided aggregation, semantic alignment, and DDI loss, and is approximately $O\big(|\mathcal{U}|\cdot L\cdot K\cdot d + |\mathcal{M}|\cdot d\big)$, where $\mathcal{M}$ is the medication set, L is the layer number, and K is the sampled neighbor number. The experiments (i.e., Section 4.7) measure the actual time cost, showing that our method is efficient.

3.6.2 ***Model Size.*** Since clients in federated learning have limited computational and storage resources [12, 23], clients should be simple and lightweight. Some existing medication recommenders [34, 43, 48, 50] can be utilized as clients, but they introduce many parameters, which increases training and communication cost. In this section, we select four representative baselines (also used in experiments) and compare their parameter sizes with our medication recommendation module. Table 1 presents the comparison results (only reporting the parameters of model architectures). For consistency, let d be the embedding size, L be the neural network layer number, $|\mathcal{M}|$ be the medication number, g be the gate number of RNN/GRU/LSTM, $|\mathcal{C}|$ be the atom token number in medications, $|\mathcal{S}|$ be the BRICS substructure number, and $\bar{s}$ the average substructure number per medication. We observe that the parameter number in our model is smaller than that of the other baseline methods.

Table 1: Model parameter size comparison.

Model	Model Parameter Size
GAMENet [34]	$(4gL + 2L + 2)d^2 + 3d\|\mathcal{M}\|$
SafeDrug [43]	$((4g + 1)L + 3)d^2 + d(\|C\| + \|\mathcal{S}\|) + \|\mathcal{M}\|\bar{s}$
RASNet [50]	$((4g + 2)L + 9)d^2 + 2d\|\mathcal{M}\| + 3d$
SSPNet [48]	$((4g + 24)L + 17)d^2 + (6L + 7)d$
Our Model	$(L + 1)d^2 + 2d$

4 Experiment

In this section, we first introduce the experimental setup and then conduct experiments to analyze the effectiveness of our FedMed.

4.1 Experimental Setup

This subsection details four aspects of the experimental setup.

4.1.1 ***Datasets.*** We adopt two real-world EHR datasets, MIMIC-III and MIMIC-IV, which have been widely used in previous studies [39, 42, 43, 48]. We first obtain the required access certificate from PhysioNet[2], then download the MIMIC-III[3] and MIMIC-IV[4] datasets, and finally preprocess the two datasets by using the pipeline in [38, 43]. The dataset statistics are summarized in Table 2.

Table 2: Statistics of the two datasets.

Items	MIMIC-III	MIMIC-IV
# of patient / clinical visit	6,350 / 15,032	50,635 / 119,176
# of diagnosis/procedure/medicine	1,958 / 1,430 / 112	2,000 / 2,419 / 118
avg. # of diagnosis per visit	10.50	7.98
avg. # of procedure per visit	3.84	2.28
avg. # of medicine per visit	11.64	6.75

4.1.2 ***Baselines.*** FedMed consists of two components: the client model and server design. Therefore, we utilize two baseline types to evaluate the effectiveness of two components. The first baseline type includes six medication recommenders for evaluating our client model, and these baselines are GAMENet [34], SafeDrug [43], 4SDrug [39], RASNet [39], ARMR [42], and SSPNet [48]. The second baseline type includes four federated learning server-side strategies for evaluating our server design, and these baselines are FedAvg [29], FedProx [24], GPFedRec [47], and CASA [26].

The descriptions of client-side baselines are listed as follows:

- GAMENet uses graph-augmented memory networks to model EHR and DDI graphs for capturing patient information [34].
- SafeDrug uses RNNs to model patient records and incorporates drug structural information when modeling medications [43].
- 4SDrug designs set-oriented representations and similarity measurements for symptoms and drugs, and introduces knowledge-based and data-driven penalties to promote safer drug sets [39].
- RASNet adopts a recurrent aggregation neural network to capture patient historical records and employs a controller on the DDI loss to enhance the safety of drug combinations.
- ARMR leverages a piecewise temporal learning design to capture dynamically changing patient conditions.
- SSPNet employs a set-based encoder to represent patient conditions and design a permutation-consistent decoder for prediction.

[2]https://physionet.org/
[3]https://physionet.org/content/mimiciii/1.4/
[4]https://physionet.org/content/mimiciv/3.1/

Table 3: Jaccard (%) and PRAUC (%) results of different client models with server update operations on MIMIC-III dataset.

Server Operations:	FedAvg (AISTATS'17)		FedProx (MLSys'20)		GPFedRec (KDD'24)		CASA (KDD'24)		Our Server Design	
Client Models:	Jaccard (↑)	PRAUC (↑)	Jaccard (↑)	PRAUC (↑)	Jaccard (↑)	PRAUC (↑)	Jaccard (↑)	PRAUC (↑)	Jaccard (↑)	PRAUC (↑)
GAMENet (AAAI'19)	44.73±0.31	65.84±0.14	45.80±0.20	65.98±0.30	46.94±0.28	67.65±0.32	46.13±0.31	67.36±0.29	47.34±0.25	68.03±0.25
SafeDrug (IJCAI'21)	45.32±0.12	67.70±0.15	46.37±0.33	67.94±0.10	45.28±0.27	68.15±0.36	45.45±0.19	67.72±0.30	47.51±0.21	68.79±0.24
4SDrug (KDD'22)	45.91±0.20	75.31±0.21	46.43±0.30	75.08±0.19	47.13±0.30	74.73±0.23	45.91±0.20	72.19±0.28	47.66±0.10	75.44±0.24
RASNet (KBS'24)	46.28±0.22	63.98±0.25	47.63±0.17	66.60±0.24	48.34±0.15	67.67±0.30	46.08±0.18	64.37±0.27	48.81±0.19	68.95±0.31
ARMR (IJCAI'25)	43.15±0.23	69.03±0.26	43.62±0.29	71.20±0.37	42.75±0.27	70.84±0.31	43.49±0.24	68.73±0.27	44.46±0.20	71.79±0.22
SSPNet (IJCAI'25)	46.12±0.25	70.13±0.35	46.04±0.16	70.81±0.29	44.31±0.19	71.35±0.20	45.23±0.23	70.16±0.17	47.73±0.18	72.75±0.26
Our Client Model	**48.63±0.20**	**76.37±0.19**	**49.54±0.18**	**76.62±0.14**	**49.12±0.21**	**76.74±0.32**	**48.90±0.15**	**76.41±0.24**	**49.63±0.17**	**77.18±0.16**

Table 4: Jaccard (%) and PRAUC (%) results of different client models with server update operations on MIMIC-IV dataset.

Server Operations:	FedAvg (AISTATS'17)		FedProx (MLSys'20)		GPFedRec (KDD'24)		CASA (KDD'24)		Our Server Design	
Client Models:	Jaccard (↑)	PRAUC (↑)	Jaccard (↑)	PRAUC (↑)	Jaccard (↑)	PRAUC (↑)	Jaccard (↑)	PRAUC (↑)	Jaccard (↑)	PRAUC (↑)
GAMENet (AAAI'19)	41.68±0.24	63.40±0.19	41.54±0.37	64.20±0.36	41.90±0.15	64.14±0.28	41.44±0.24	63.57±0.29	42.14±0.20	64.56±0.36
SafeDrug (IJCAI'21)	42.74±0.13	68.14±0.20	43.10±0.17	68.57±0.18	43.02±0.32	68.96±0.21	41.85±0.27	67.50±0.34	43.35±0.19	69.44±0.26
4SDrug (KDD'22)	44.19±0.29	70.87±0.22	43.59±0.26	70.68±0.39	44.28±0.18	71.25±0.33	43.30±0.18	69.02±0.25	44.75±0.31	71.27±0.10
RASNet (KBS'24)	45.15±0.34	63.46±0.33	44.81±0.25	64.18±0.19	45.07±0.16	65.02±0.28	44.92±0.30	61.77±0.21	45.29±0.12	65.23±0.22
ARMR (IJCAI'25)	42.20±0.21	64.44±0.32	42.08±0.20	63.99±0.25	42.34±0.13	64.31±0.14	42.86±0.32	65.06±0.36	43.11±0.27	65.28±0.29
SSPNet (IJCAI'25)	44.05±0.27	68.25±0.29	43.77±0.23	67.71±0.33	41.99±0.22	68.27±0.36	42.63±0.17	68.05±0.25	44.48±0.18	69.97±0.20
Our Client Model	**45.76±0.25**	**72.14±0.12**	**46.09±0.19**	**72.78±0.27**	**45.43±0.20**	**73.30±0.26**	**46.20±0.17**	**72.57±0.24**	**46.69±0.08**	**73.58±0.19**

Table 5: DDI (%) results of different client models with server update operations on MIMIC-III and MIMIC-IV datasets.

Server Operations:	FedAvg (AISTATS'17)		FedProx (MLSys'20)		GPFedRec (KDD'24)		CASA (KDD'24)		Our Server Design	
Client Models:	MIMIC-III DDI (↓)	MIMIC-IV DDI (↓)	MIMIC-III DDI (↓)	MIMIC-IV DDI (↓)	MIMIC-III DDI (↓)	MIMIC-IV DDI (↓)	MIMIC-III DDI (↓)	MIMIC-IV DDI (↓)	MIMIC-III DDI (↓)	MIMIC-IV DDI (↓)
GAMENet (AAAI'19)	10.98±0.07	12.87±0.07	10.73±0.08	12.76±0.14	10.19±0.07	13.03±0.13	10.50±0.09	13.20±0.09	10.06±0.14	12.19±0.10
SafeDrug (IJCAI'21)	9.31±0.16	12.80±0.14	9.50±0.17	12.87±0.09	9.67±0.19	12.24±0.16	9.45±0.08	12.95±0.11	9.18±0.12	11.78±0.17
4SDrug (KDD'22)	9.68±0.13	12.21±0.13	9.54±0.22	**12.30±0.12**	9.85±0.20	11.60±0.18	9.12±0.14	11.94±0.15	8.87±0.11	11.17±0.13
RASNet (KBS'24)	9.22±0.09	12.59±0.15	9.78±0.21	12.70±0.07	9.16±0.12	12.83±0.18	9.09±0.10	12.46±0.16	8.95±0.18	11.90±0.08
ARMR (IJCAI'25)	9.71±0.15	13.12±0.16	9.95±0.20	12.65±0.19	9.82±0.18	12.94±0.15	9.44±0.13	13.31±0.21	8.97±0.14	12.58±0.15
SSPNet (IJCAI'25)	10.26±0.15	13.61±0.15	10.14±0.12	13.54±0.16	9.49±0.07	13.85±0.10	9.68±0.17	13.58±0.13	9.25±0.17	13.42±0.10
Our Client Model	**9.03±0.11**	**11.67±0.09**	**8.69±0.09**	12.46±0.12	**9.10±0.13**	**11.37±0.05**	**8.94±0.14**	**11.82±0.19**	**8.50±0.05**	**10.71±0.09**

The descriptions of server-side baselines are listed as follows:

- FedAvg updates the parameters uploaded from clients on the server by averaging them [29].
- FedProx extends FedAvg by adding a proximal term to improve training stability and convergence [24].
- GPFedRec constructs a client relationship graph on the server based on uploaded parameters to guide parameter updates [47].
- CASA uses asynchronous aggregation and dynamic clustering with a buffer to coordinate clustering and parameter update [26].

4.1.3 ***Hyper-parameter Settings.*** For baselines, we either use their source code (i.e., GAMENet, SafeDrug, 4SDrug, RASNet, ARMR, FedAvg, FedProx, and GPFedRec) or implement them ourselves (i.e., SSPNet and CASA) when the code is not publicly available. The search spaces for baseline hyperparameters are as follows: the learning rate is selected in $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. We search embedding size d in $\{8, 16, 32, 64, 128\}$. The layer sizes of neural networks (e.g., the multi-layer perceptron in [48, 50] and GNN in [34, 43, 48, 50]) are selected from $\{1, 2, 3, 4\}$. For the DDI loss strength in [34, 39, 43, 50] is adjusted from $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. Other hyperparameter settings of baselines are researched by empirical study or following original papers. For our FedMed, on the client side, we set the embedding size to 16, contrastive learning and DDI constraint strengths are selected from $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$, and privacy-preserving noise intensity is chosen from $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$. On the server side, layer and sampled neighbor numbers are selected from $\{1, 2, 3, 4\}$ and $\{5, 10, 15, 20, 25, 30\}$. These hyperparameters are carefully tuned, and a detailed analysis is provided in Sections 4.5 and 4.6.

4.1.4 ***Evaluation Protocols.*** We evaluate medication recommendation performance using metrics from accuracy and safety [39, 42, 43, 48]. For accuracy, we use Jaccard and PRAUC, where Jaccard quantifies the overlap between the predicted and ground-truth medication sets, and PRAUC evaluates overall prediction quality based on the precision recall curve. Higher values of both metrics indicate better accuracy. For safety, we utilize the DDI rate, which measures the risk of harmful drug interactions. A lower DDI rate indicates better safety. In our experiments, we adopt leave-one-out evaluation, which has been widely used in federated recommender systems [46, 47]. Specifically, for each patient, we use the last clinical record as the test set, the second-to-last record as the validation set, and all others as the training set.

4.2 Performance Comparison

The comparative results on the two datasets are shown in Tables 3, 4, and 5. Specifically, Tables 3 and 4 report the accuracy results for medication recommendation, while Table 5 presents the safety results. Our FedMed consists of two key components (i.e., the client and server), and we next analyze both components separately.

From the client-design perspective, Tables 3, 4, and 5 compare our model with various recommenders baselines. These baselines

often achieve nice performance in centralized storage settings. However, when deployed as client models in federated learning, their performance becomes less satisfactory. This may be because their complex architectures tend to overfit the sparse patient data on each client. Moreover, such complexity is not well-suited to client devices with limited computational resources. In terms of accuracy, our client achieves the best performance, which can be attributed to the simple yet effective attention-based multi-pooling and self-supervised strategy. Regarding safety, our client also performs best in most cases, benefiting from the incorporation of the DDI loss.

From the server-design perspective, Tables 3, 4, and 5 compare our server architecture with those of other federated learning methods. Results show that our methods generally achieve the best performance in accuracy and safety, which can be attributed to the graph-guided aggregation, semantic alignment operation, and DDI loss, all of which are tailored to medication recommendation. In contrast, the other server-design baselines mainly adopt vanilla federated parameter update strategies and lack task-specific designs.

4.3 Performance w.r.t Sparsity Degrees

We analyze how our method performs under sparse scenarios, and train it on only 75%, 50%, and 25% of the training data from patient records, while keeping the validation and test sets unchanged. We also compare our client design against three representative baselines (i.e., GAMENet, RASNet, and SSPNet), where all models are evaluated under the same server setting (i.e., our server design). The results are shown in Figure 2 and we can observe that: (1) as the training data decreases, the performance of all models drops, which shows that sufficient data is important for medication recommendation, and (2) our FedMed performs best under all training data ratios and shows the smallest drop. For example, on the MIMIC-III dataset with only 25% training data, the performance of FedMed drops by only 5.208%, while GAMENet, RASNet, and SSPNet drop by 9.643%, 9.332%, and 7.165%, respectively. These results show the robustness and effectiveness of our FedMed in sparse scenarios.

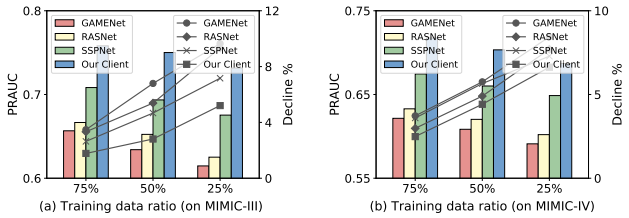


Figure 2: Model performance w.r.t. training data ratio. The bar represents PRAUC results, and lines represent the relative performance drop compared to the full training set.

4.4 Ablation Study

The client and server modules are two crucial components of our FedMed. To analyze the client, we consider three operations:

- w/o ATT: removing the ATTention mechanism.
- w/o SSL: removing the Self-Supervised Learning strategy.
- w/o DDI(C): removing the Client-side DDI loss.

For the server module, we consider three operations:

- w/o AGG: removing the graph-guided AGGregation.
- w/o SA: removing the Semantic Alignment module.
- w/o DDI(S): removing the Server-side DDI loss.

Table 6 shows the results for PRAUC and DDI on the two datasets. We can have the following observations: (1) removing ATT, SSL, AGG, and SA operations leads to lower PRAUC compared with the full FedMed model, suggesting that these operations contribute to recommendation accuracy, while their effect on the safety metric (DDI rate) is marginal, and (2) removing DDI(C) or DDI(S) increases the DDI rate but yields higher accuracy. This is because removing the DDI loss reduces the constraints among medications (e.g., allowing more medications to be included in a single recommendation combination), which enlarges the candidate space and helps improve predictive accuracy. The hyper-parameter study in Section 4.5 further shows the impact of the DDI loss weights.

Table 6: Ablation study of key designs in our FedMed.

Operation	MIMIC-III		MIMIC-IV	
	PRAUC (%)	DDI (%)	PRAUC (%)	DDI (%)
w/o ATT	77.03±0.18	8.51±0.06	73.26±0.20	10.74±0.09
w/o SSL	76.29±0.13	8.54±0.09	71.80±0.18	10.57±0.05
w/o DDI(C)	77.70±0.21	9.17±0.07	73.94±0.13	11.96±0.11
w/o AGG	76.15±0.14	8.62±0.12	71.57±0.15	10.79±0.08
w/o SA	76.66±0.17	8.75±0.10	72.40±0.21	10.89±0.06
w/o DDI(S)	77.24±0.15	9.01±0.08	73.65±0.16	11.18±0.12
FedMed	77.18±0.16	8.50±0.05	73.58±0.19	10.71±0.09

4.5 Hyper-parameter Sensitivity Analysis

We analyze four hyperparameters of our FedMed, including two client-side parameters (i.e., DDI loss weight λ^c_{DDI} and self-supervised learning weight λ_{SSL}) and two server-side parameters (i.e., sampled neighbor number K and aggregation layer number L). The results on MIMIC-III dataset are shown in Figure 3, and similar trends are observed on MIMIC-IV dataset. We can find that (1) for the DDI loss weight, as discussed in Section 4.4, it balances medication safety and recommendation accuracy. Setting it too large reduces the DDI rate but degrades recommendation accuracy. Therefore, it provides a practical way for clinicians to control the trade-off between medication safety and recommendation accuracy; (2) for the self-supervised learning weight, setting it too small provides insufficient constraints for data sparsity mitigation, while setting it too large shifts the training focus away from recommendation accuracy, leading to degraded performance; and (3) for the sampled neighbor and aggregation layer numbers, larger values bring limited accuracy performance gains but introduce higher computational cost. In practice, setting K to 10 or 15 and L to 1 provides a good trade-off between effectiveness and efficiency.

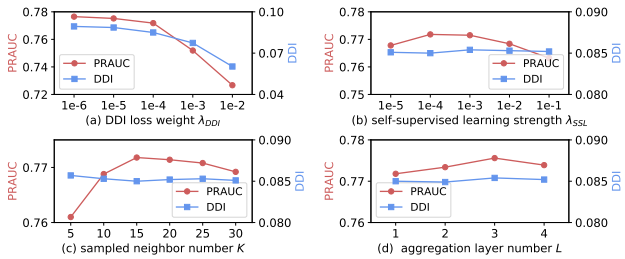


Figure 3: Hyperparameter sensitivity on MIMIC-III dataset.

4.6 Privacy Protection Analysis

Our FedMed is a medication recommendation model developed under a federated learning setting. Due to the decentralized training architecture of federated learning, raw patient data remains local, which naturally provides a certain level of privacy protection. To further enhance the security of parameter upload, we incorporate a differential privacy technique (i.e., Section 3.2). Specifically, we add Laplacian noise during client updates, where a larger noise strength δ (in Eq.(11)) provides stronger privacy protection. This section analyzes the impact of noise strength δ on our FedMed performance by varying δ from 0.1 to 0.8 with a step size of 0.1. The results are shown in Table 7. We observe that as the noise strength increases, the accuracy metric (PRAUC) gradually decreases, while the safety metric (DDI rate) remains relatively stable. This indicates a trade-off between privacy and accuracy. Based on this observation, we recommend setting δ to 0.1 or 0.2 as a practical balance.

Table 7: Noise intensity analysis on the MIMIC-III dataset.

Intensity	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
PRAUC (%)	77.18	77.10	76.80	76.28	75.51	75.47	75.50	75.36
DDI (%)	8.50	8.47	8.56	8.51	8.53	8.45	8.44	8.49

4.7 Time Efficiency Analysis

We analyze the model computational cost from both the client and server sides. For the client-side comparison, we deploy our client design and all baselines under the same server update setting (e.g., our server design). For the server-side comparison, we fix our client design and evaluate different server update strategies. Figure 4 shows the training time cost per epoch and testing inference time. We observe that: (1) in the client-side comparison, our client design achieves a lower time cost, showing its lightweight and efficient architecture; (2) in the server-side comparison, our method introduces slightly higher overhead due to the graph-guided aggregation, semantic alignment, and DDI loss. However, these operations improve both the accuracy and safety of medication recommendations. Overall, the server-side overhead remains on the same order of magnitude as other methods; and (3) generally speaking, the training and testing time of our FedMed is acceptable in practice.

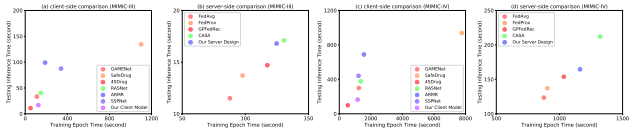


Figure 4: Time efficiency analysis of the client and server.

5 Related Work

In this section, we review two research areas related to our work.

5.1 Medication Recommendation

Automated medication recommendation is a crucial task in AI for healthcare [2, 49]. Early methods encode clinician expertise as recommendation rules [3, 14], but they require heavy expert effort and often transfer poorly to new hospitals or patient groups. With the growth of deep learning, researchers use neural networks (e.g., RNN, GNN, and Transformer) to model clinical records and learn prescribing patterns from data, which reduces the reliance on hand-crafted rules and improves recommendation quality [25, 34, 42–44, 48, 50]. However, these methods typically rely on centralized storage of patient records to train a centrally managed recommendation model, which may expose patient sensitive information. To this end, in this paper, we propose FedMed, a privacy-preserving, personalized, and safe medication recommender based on federated learning. To the best of our knowledge, FedMed is among the early efforts to apply federated learning to the medication recommendation task.

5.2 Federated Recommendation

Recently, federated learning [22] has been widely used in recommendation, as recommender tasks naturally involve sensitive user information and require privacy protection [11, 35]. Generally, each user is treated as a client that trains a local model using only private data, and a server coordinates the training process by aggregating client updates and distributing global parameters [10, 28, 30, 36, 45]. In this paper, we study how to deploy medication recommendation in federated settings. Due to several task-specific challenges (e.g., sparse patient records and DDI safety constraints), existing federated recommenders cannot be directly used for medication recommendation. To this end, we propose FedMed with client-side design (e.g., self-supervised strategy) and server-side design (e.g., DDI loss) tailored to this task. Our model supports personalization, preserves privacy, and promotes safe medication combinations.

6 Conclusion

In this paper, we propose a medication recommender built on the federated learning paradigm. On the client side, we design an effective prediction model that generates medication prediction based on each patient private clinical records. To better learn from sparse records, we further introduce a self-supervised learning strategy to provide additional training signals. On the server side, we construct a patient relationship graph without exposing sensitive information, and develop a graph-guided aggregation mechanism to improve the client updating and synchronization. In addition, to reduce harmful adverse interactions in recommended medication sets, we incorporate DDI regularization on both the client and server sides. We conduct extensive experiments on two benchmark datasets, and the results demonstrate the effectiveness of our FedMed.

Limitations and Ethical Considerations

We use de-identified ICU EHR data (i.e., MIMIC-III and MIMIC-IV datasets) under the data use agreement. No new data was collected and no re-identification was attempted. We follow relevant ACM policies and requirements. Privacy and consent risks are low. Limitations include ICU-only data, and missing and noisy records.

Acknowledgments

The authors also thank the anonymous reviewers for their helpful comments. This research has in part been funded by the Research Council of Finland (grant 339421) and Technology Industries of Finland Centennial Foundation (Aalto University House of AI). Part of this work was also funded by the National Plan for NRRP Complementary Investments (PNC) in the call for the funding of research initiatives for technologies and innovative trajectories in the health-project n. PNC0000003-AdvaNced Technologies for Human-centrEd Medicine (project acronym: ANTHEM).

References

[1] Walid Ahmad, Elana Simon, Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. 2022. Chemberta-2: Towards chemical foundation models. *arXiv preprint arXiv:2209.01712* (2022).

[2] Zafar Ali, Yi Huang, Irfan Ullah, Junlan Feng, Chao Deng, Nimbeshaho Thierry, Asad Khan, Asim Ullah Jan, Xiaoli Shen, Wu Rui, et al. 2023. Deep learning for medication recommendation: a systematic survey. *Data Intelligence* 5, 2 (2023), 303–354.

[3] Daniel Almirall, Scott N Compton, Meredith Gunlicks-Stoessel, Naihua Duan, and Susan A Murphy. 2012. Designing a pilot sequential multiple assignment randomized trial for developing an adaptive treatment strategy. *Statistics in medicine* 31, 17 (2012), 1887–1902.

[4] Maryam Astero, Anchen Li, Elena Casiraghi, and Juho Rousu. 2026. A cross-attentive multi-task graph learning framework for chemical reaction modeling. *Bioinformatics* 42, 5 (2026), btag193.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*. PmLR, 1597–1607.

[6] Woo-Seok Choi, Matthew Tomei, Jose Rodrigo Sanchez Vicarte, Pavan Kumar Hanumolu, and Rakesh Kumar. 2018. Guaranteeing local differential privacy on ultra-low-power systems. In *ISCA*. IEEE, 561–574.

[7] Meredith Gunlicks-Stoessel, Laura Mufson, Ana Westervelt, Daniel Almirall, and Susan Murphy. 2016. A pilot SMART for developing an adaptive treatment strategy for adolescent depression. *Journal of Clinical Child & Adolescent Psychology* 45, 4 (2016), 480–494.

[8] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*. JMLR Workshop and Conference Proceedings, 297–304.

[9] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[10] Xinrui He, Shuo Liu, Jacky Keung, and Jingrui He. 2024. Co-clustering for federated recommender system. In *WWW*. 3821–3832.

[11] Danish Javeed, Muhammad Shahid Saeed, Prabhat Kumar, Alireza Jolfaei, Shareeful Islam, and AKM Najmul Islam. 2023. Federated learning-based personalized recommendation systems: An overview on security and privacy challenges. *IEEE Transactions on Consumer Electronics* 70, 1 (2023), 2618–2627.

[12] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurelien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and trends in machine learning* 14, 1–2 (2021), 1–210.

[13] Rayan Krishnan, Pranav Rajpurkar, and Eric J Topol. 2022. Self-supervised learning in medicine and healthcare. *Nature Biomedical Engineering* 6, 12 (2022), 1346–1352.

[14] Himabindu Lakkaraju and Cynthia Rudin. 2017. Learning cost-effective and interpretable treatment regimes. In *Artificial intelligence and statistics*. PMLR, 166–175.

[15] Anchen Li, Elena Casiraghi, and Juho Rousu. 2024. Chemical reaction enhanced graph learning for molecule representation. *Bioinformatics* 40, 10 (2024), btae558.

[16] Anchen Li, Elena Casiraghi, and Juho Rousu. 2025. CSGL: chemical synthesis graph learning for molecule representation. *Bioinformatics* 41, 7 (2025), btaf355.

[17] Anchen Li and Bo Yang. 2025. Dual graph denoising model for social recommendation. In *Proceedings of the ACM on Web Conference 2025*. 347–356.

[18] Anchen Li, Bo Yang, Huan Huo, and Farookh Hussain. 2022. Hypercomplex graph collaborative filtering. In *Proceedings of the ACM Web Conference 2022*. 1914–1922.

[19] Anchen Li, Bo Yang, Huan Huo, Farookh Hussain, and Guandong Xu. 2025. Hypercomplex knowledge graph-aware recommendation. In *Proceedings of the 48th international ACM SIGIR conference on research and development in information retrieval*. 2017–2026.

[20] Anchen Li, Bo Yang, Huan Huo, Farookh Khadeer Hussain, and Guandong Xu. 2024. Structure-and logic-aware heterogeneous graph learning for recommendation. In *2024 IEEE 40th international conference on data engineering (ICDE)*. IEEE, 544–556.

[21] Anchen Li, Bo Yang, Huan Huo, Farookh Khadeer Hussain, and Guandong Xu. 2025. Self-supervised dual graph learning for recommendation. *Knowledge-Based Systems* 310 (2025), 112967.

[22] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. 2020. A review of applications in federated learning. *Computers & Industrial Engineering* 149 (2020), 106854.

[23] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine* 37, 3 (2020), 50–60.

[24] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *MLSys* 2 (2020), 429–450.

[25] Xiang Li, Shunpan Liang, Yu Lei, Chen Li, Yulei Hou, Dashun Zheng, and Tengfei Ma. 2024. Causalmed: Causality-based personalized medication recommendation centered on patient health state. In *CIKM*. 1276–1285.

[26] Boyi Liu, Yiming Ma, Zimu Zhou, Yexuan Shi, Shuyuan Li, and Yongxin Tong. 2024. Casa: Clustered federated learning with asynchronous clients. In *KDD*. 1851–1862.

[27] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *IEEE transactions on knowledge and data engineering* 35, 1 (2021), 857–876.

[28] Zhiwei Liu, Liangwei Yang, Ziwei Fan, Hao Peng, and Philip S Yu. 2022. Federated social recommendation with graph neural network. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13, 4 (2022), 1–24.

[29] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*. PMLR, 1273–1282.

[30] Wu Meihan, Li Li, Chang Tao, Eric Rigall, Wang Xiaodong, and Xu Cheng-Zhong. 2022. Fedcdr: federated cross-domain recommendation for privacy-preserving rating prediction. In *CIKM*. 2179–2188.

[31] Maria Panagioti, Jonathan Stokes, Aneez Esmail, Peter Coventry, Sudeh Cheraghi-Sohi, Rahul Alam, and Peter Bower. 2015. Multimorbidity and patient safety incidents in primary care: a systematic review and meta-analysis. *PloS one* 10, 8 (2015), e0135947.

[32] Formerly Data Protection. 2018. General data protection regulation (GDPR). *Intersoft Consulting, Accessed in October* 24, 1 (2018).

[33] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. *NIPS* 30 (2017).

[34] Junyuan Shang, Cao Xiao, Tengfei Ma, Hongyan Li, and Jimeng Sun. 2019. Gamenet: Graph augmented memory networks for recommending medication combination. In *AAAI*, Vol. 33. 1126–1133.

[35] Linrui Shen, Anchen Li, Xueyan Liu, Riting Xia, and Bo Yang. 2026. Dual-Channel Personalized Federated Bundle Recommendation. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1146–1150.

[36] Linrui Shen, Anchen Li, Xueyan Liu, Riting Xia, and Bo Yang. 2026. Grouping-enhanced personalization for federated recommendation. *Information Processing & Management* 63, 4 (2026), 104599.

[37] Benjamin Stark, Constanze Knahl, Mert Aydin, and Karim Elish. 2019. A literature review on medicine recommender systems. *International journal of advanced computer science and applications* 10, 8 (2019).

[38] Hongda Sun, Shufang Xie, Shuqi Li, Yuhan Chen, Ji-Rong Wen, and Rui Yan. 2022. Debiased, longitudinal and coordinated drug recommendation through multi-visit clinic records. *NIPS* 35 (2022), 27837–27849.

[39] Yanchao Tan, Chengjun Kong, Leisheng Yu, Pan Li, Chaochao Chen, Xiaolin Zheng, Vicki S Hertzberg, and Carl Yang. 2022. 4SDrug: Symptom-based Set-to-set Small and Safe Drug Recommendation. In *KDD*. 3970–3980.

[40] Marjan Van Den Akker, Frank Buntinx, and J André Knottnerus. 1996. Comorbidity or multimorbidity: what's in a name? A review of literature. *The European journal of general practice* 2, 2 (1996), 65–70.

[41] David Weininger. 1988. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences* 28, 1 (1988), 31–36.

[42] Feiyue Wu, Tianxing Wu, and Shenqi Jing. 2025. ARMR: Adaptively Responsive Network for Medication Recommendation. In *IJCAI*. 7831–7839.

[43] Chaoqi Yang, Cao Xiao, Fenglong Ma, Lucas Glass, and Jimeng Sun. 2021. SafeDrug: Dual Molecular Graph Encoders for Recommending Effective and Safe Drug Combinations. In *IJCAI*. 3735–3741.

[44] Nianzu Yang, Kaipeng Zeng, Qitian Wu, and Junchi Yan. 2023. Molerec: Combinatorial drug recommendation with substructure-aware molecular representation learning. In *WWW*. 4075–4085.

[45] Jingwei Yi, Fangzhao Wu, Chuhan Wu, Ruixuan Liu, Guangzhong Sun, and Xing Xie. 2021. Efficient-FedRec: Efficient federated learning framework for privacy-preserving news recommendation. *arXiv preprint arXiv:2109.05446* (2021).

[46] Chunxu Zhang, Guodong Long, Tianyi Zhou, Peng Yan, Zijian Zhang, Chengqi Zhang, and Bo Yang. 2023. Dual personalization on federated recommendation. In *IJCAI*. 4558–4566.

[47] Chunxu Zhang, Guodong Long, Tianyi Zhou, Zijian Zhang, Peng Yan, and Bo Yang. 2024. Gpfedrec: Graph-guided personalization for federated recommendation. In *KDD*. 4131–4142.

[48] Haodi Zhang, Jiawei Wen, Jiahong Li, Yuanfeng Song, Liang-Jie Zhang, and Lin Ma. 2025. SSPNet: Leveraging Robust Medication Recommendation with History and Knowledge. In *IJCAI*. 9465–9473.

[49] Fanglin Zhu, Lizhen Cui, Yonghui Xu, Zhe Qu, and Zhiqi Shen. 2024. A survey of personalized medicine recommendation. *International Journal of Crowd Science* 8, 2 (2024), 77–82.

[50] Qiang Zhu, Feng Han, Huali Yang, Junping Liu, Xinrong Hu, and Bangchao Wang. 2024. RASNet: Recurrent aggregation neural network for safe and efficient drug recommendation. *Knowledge-Based Systems* 299 (2024), 112055.