FedSLLM: LLM-Derived Semantic Prototypes for Sample Selection in Federated Recommendation

Anonymous ACL submission

Abstract

Recently, large language models (LLMs) have demonstrated strong generative capabilities. These advances create new opportunities for improving federated recommendation (FR), which enables distributed model training while preserving user privacy. However, strict constraints on privacy, fairness, and communication overhead leave a research gap in applying LLMs to FR, particularly in addressing ineffective training caused by biased or unrepresentative samples. To this end, we propose FedSLLM, an FR framework leveraging server-side LLMs to generate semantic prototypes that guide clients in selecting the most informative and representative local samples based on semantic relevance and prediction difficulty. This approach enables effective, lightweight, and privacy-preserving sample selection without deploying LLMs on clients or sharing raw data. Extensive experiments on multiple FR backbones and datasets show that FedSLLM consistently improves recommendation performance, especially under low sampling ratios, while reducing the amount of training data required. Our code is available at https://anonymous.4open.science/r/fl_s-AD54.

1 Introduction

Large language models (LLMs) have demonstrated a strong capability to capture rich semantic information, which can guide more effective learning across various stages of model training. In federated recommendation (FR) (McMahan et al., 2017; Yang et al., 2020; Yuan et al., 2025a), a privacy-preserving distributed learning paradigm where user interaction data remain local due to regulations such as the General Data Protection Regulation (GDPR) (Voigt and Von dem Bussche, 2017), leveraging such semantic guidance is particularly promising. By incorporating high-level semantic knowledge, FR systems can better compensate for data heterogeneity across clients, thereby improving recommendation (Zhao et al., 2025; Yuan et al., 2025b). However, two challenges hinder the effective and efficient use of LLMs in FR:

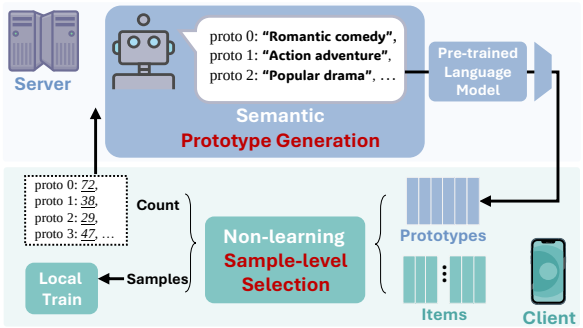


Figure 1: LLM-generated semantic prototypes for sample-level selection in federated recommendation.

(1) Mitigating client bias. Each client observes only a limited and potentially biased subset of interactions, which can lead to unrepresentative local training. In conventional federated learning (FL), auxiliary models and additional information have been used to select more important clients (Pan et al., 2025; Wang et al., 2025a). Extending this idea, a natural research direction is to leverage LLMs as auxiliary models to improve client-level selection, taking advantage of their rich semantic knowledge. However, in FR, excluding clients is inappropriate because it violates fairness and reduces coverage of diverse user behaviors (Mukhtiar et al., 2025). To address this issue, we propose LLM-enhanced sample-level selection, where LLMs guide clients to select the most informative and representative local samples, mitigating bias without excluding any client.

(2) Providing semantic guidance efficiently. Prior works have explored the idea of combining LLMs with sample-level selection in centralized settings (Tian et al., 2025; Tong et al., 2025), but directly deploying LLMs on resource-constrained clients is impractical due to computation overhead. Additionally, privacy regulations prevent sharing

raw interactions or semantic signals with the server, which limits naive LLM-based approaches. Our framework overcomes these challenges by generating semantic prototypes at the server and distributing only lightweight guidance to clients, enabling them to leverage semantic information efficiently while respecting privacy constraints.

To this end, we propose FedSLLM, an LLM-enhanced prototype-based sample selection framework for FR. As shown in Figure 1, the server uses an LLM to generate a compact set of semantic prototypes, which guide clients in selecting representative local samples.

Specifically, FedSLLM follows a cluster-based sampling principle, where prototypes serve as cluster centers and samples are chosen based on similarity (Morafah et al., 2023; Fraboni et al., 2021; Song et al., 2023). Unlike data-driven or learned clusters, LLM-generated prototypes incorporate semantic knowledge without accessing private interactions. Furthermore, FedSLLM employs a non-learning hard-sample selection strategy, selecting samples semantically close to prototypes but with divergent predicted outcomes, thus focusing training on challenging cases at minimal extra cost (Shi et al., 2023; Lee et al., 2025). Finally, clients in FedSLLM upload lightweight feedback on the prototypes, which does not disclose any privacy to the server. The server then aggregates this feedback to update the prototypes, making them more representative and better semantically aligned with the interaction space. In a nutshell, the key contributions of this study are summarized as follows:

- We identify LLMs as a powerful yet largely unexplored source of semantic guidance for federated recommendation (FR), and present the first study that investigates how LLM-driven sample-level selection can improve the performance-efficiency trade-off.
- We propose FedSLLM, a prototype-based framework designed for LLM guidance, which integrates LLM-based semantic prototype generation, prototype-aware sampling, and lightweight feedback, enabling effective and privacy-preserving sample selection.
- We align user-item interaction information with high-level semantic knowledge, including textual item descriptions and LLM-generated semantic prototypes in a privacy-preserving way, within the FR setting.

2 Related Work

LLM-enhanced recommendation. Recent studies have explored large language models (LLMs) in recommendation because of their strong generation capabilities. Existing works primarily leverage LLMs to enrich interaction representations (Zhang et al., 2025a), generate auxiliary natural language features (Wang et al., 2025b; Xu et al., 2025b), or support downstream objectives (Li et al., 2025; Wang et al., 2023a). However, directly deploying LLMs on clients is impractical in real-world federated recommendation (FR) due to their high computational and memory requirements.

In this work, we aim to leverage LLMs in federated recommendation (FR), which have demonstrated great potential in improving performance, while avoiding direct deployment on clients.

Data heterogeneity in FR. An important challenge in FR stems from the inherent data heterogeneity. Client-level selection is a common strategy in federated learning (FL) to mitigate heterogeneity and reduce training on low-value clients (Wang et al., 2023b; Chen and Vikalo, 2024; Pan et al., 2025). Representative approaches leverage auxiliary agents identify informative clients (Zhang et al., 2022; Wang et al., 2025a).

However, in FR, each client represents a unique user. Excluding clients from training can increase unfairness and harm personalization, limiting the applicability of conventional client-level selection.

Sample-level selection. To overcome the limitations discussed above, recent studies have investigated sample-level selection (Qi et al., 2023; Xu et al., 2025a), relying on auxiliary models and information to estimate sample importance. For example, Zhang et al. (2025b) introduce a meta-model trained on a proxy dataset to guide local sampling, while federated active learning approaches select samples based on aggregating sample information (Zong et al., 2025; Tang et al., 2025). Despite their effectiveness in FL, these methods face critical challenges in FR. Aggregating interaction at the server raises privacy concerns, while auxiliary models incur non-negligible computational overhead on resource-constrained clients.

In contrast, we propose a prototype-based sample selection framework tailored for FR. By leveraging LLM-generated textual prototypes as semantic anchors, our method enables efficient and privacy-preserving sample selection.

3 Preliminaries

In federated recommendation (FR), a central server maintains an item set $\mathcal{I} = \{i\}_{i=1}^{n}$, where each item i is associated with the textual attribute $a_i \in \mathcal{A}$. There are m clients $\mathcal{U} = \{u\}_{u=1}^{m}$, each corresponding to a unique user. Client u locally stores its private interaction data $D_u = \{(i, r_{ui}) \mid i \in \mathcal{I}\}$, where $r_{ui} \in \{0, 1\}$ denotes the implicit feedback indicating whether user u has interacted with item i.

In each FR round t, the server samples a subset of clients $U_t \subseteq \mathcal{U}$ according to a participation ratio $\mathcal{C}$, and broadcasts the global model $\Theta_t = \{E_t, W_g^t\}$, where $E_t = \{e_i^t \in \mathbb{R}^k\}_{i=1}^{n}$ is the item embedding table and W_g^t denotes the shared or personalized recommendation model. Each client $u \in U_t$ performs local optimization on D_u and uploads model updates to the server, which aggregates them to obtain Θ_{t+1}. After convergence, the final global model is used for local inference.

This work focuses on sample-level selection in FR. Specifically, before local training in each round, client u selects a subset

$$\tilde{D}_u \subseteq D_u, \quad |\tilde{D}_u| = \mathcal{S}\,|D_u|, \tag{1}$$

where $\mathcal{S}$ is the sampling ratio. Only samples in $\tilde{D}_u$ are used for local model updates.

4 Methodology

As shown in Figure 2, our method leverages a large language model (LLM) to generate semantic prototypes that guide sampling (Sec 4.2). Clients select informative samples based on semantic-similarity and prediction-diversity to the prototypes (Sec 4.3) and provide lightweight feedback (Sec 4.4). After local training, the server aggregates the feedback to update the prototypes and maps them into the same latent space with item embeddings by the semantic encoder (Sec 4.5).

4.1 Pre-training with Semantics

To incorporate semantic information into FR, we perform a server-side pre-training stage before federated training. Each item $i \in \mathcal{I}$ is associated with a natural language description or attribute set $a_i \in \mathcal{A}$. A pre-trained language model (PLM), denoted as $\mathcal{F}_{\text{PLM}}$, is used to encode item attributes into semantic representations:

$$x_i = \mathcal{F}_{\text{PLM}}(a_i), \quad x_i \in \mathbb{R}^{d_{\text{plm}}}. \tag{2}$$

To align the PLM representations with the embedding space used in FR, we further employ an autoencoder (AE) composed of an encoder $\mathcal{E}_{\text{AE}}$ and a decoder $\mathcal{D}_{\text{AE}}$. The encoder projects the PLM embedding into a low-dimensional latent space:

$$z_i = \mathcal{E}_{\text{AE}}(x_i), \quad z_i \in \mathbb{R}^d, \tag{3}$$

and the decoder reconstructs the representation:

$$\hat{x}_i = \mathcal{D}_{\text{AE}}(z_i). \tag{4}$$

The autoencoder is trained by minimizing the reconstruction loss:

$$\mathcal{L}_{\text{AE}} = \sum_{i \in \mathcal{I}} \|x_i - \hat{x}_i\|_2^2 . \tag{5}$$

After convergence, the latent representations $\{z_i\}_{i \in \mathcal{I}}$ are used to initialize the global item embedding table E in FR:

$$e_i^{(0)} = z_i, \quad \forall i \in \mathcal{I}, \tag{6}$$

where $e_i^{(0)}$ denotes the initial item embedding at the beginning of federated training. All pre-training procedures are conducted exclusively on the server and do not involve any client-side interaction data.

4.2 Semantic Prototype Generation

At the beginning of each federated training round t, the server queries an LLM to generate a compact set of textual prototypes, denoted as $\mathcal{P}_t = \{p_j\}_{j=1}^{N_t}$.

To ensure representation consistency between prototypes and items, the server encodes each prototype using the same PLM $\mathcal{F}_{\text{PLM}}$ employed in the pre-training stage, followed by the autoencoder encoder $\mathcal{E}_{\text{AE}}$. Specifically, the latent embedding of prototype p_j is obtained as

$$e_j^p = \mathcal{E}_{\text{AE}}\big(\mathcal{F}_{\text{PLM}}(p_j)\big), \quad e_j^p \in \mathbb{R}^d. \tag{7}$$

The resulting prototype embedding set is denoted as $E_t^p = \{e_j^p\}_{j=1}^{N_t}$.

To support semantic-aware sample selection on clients, the server computes the similarity between each item embedding $e_i \in E$ and each prototype embedding e_j^p. We adopt cosine similarity and define the similarity score as

$$\text{sim}(i, p_j) = \frac{e_i^\top e_j^p}{\|e_i\|_2 \, \|e_j^p\|_2}. \tag{8}$$

The server then distributes the prototype embeddings E_t^p along with the corresponding similarity scores to the selected clients for local sampling.

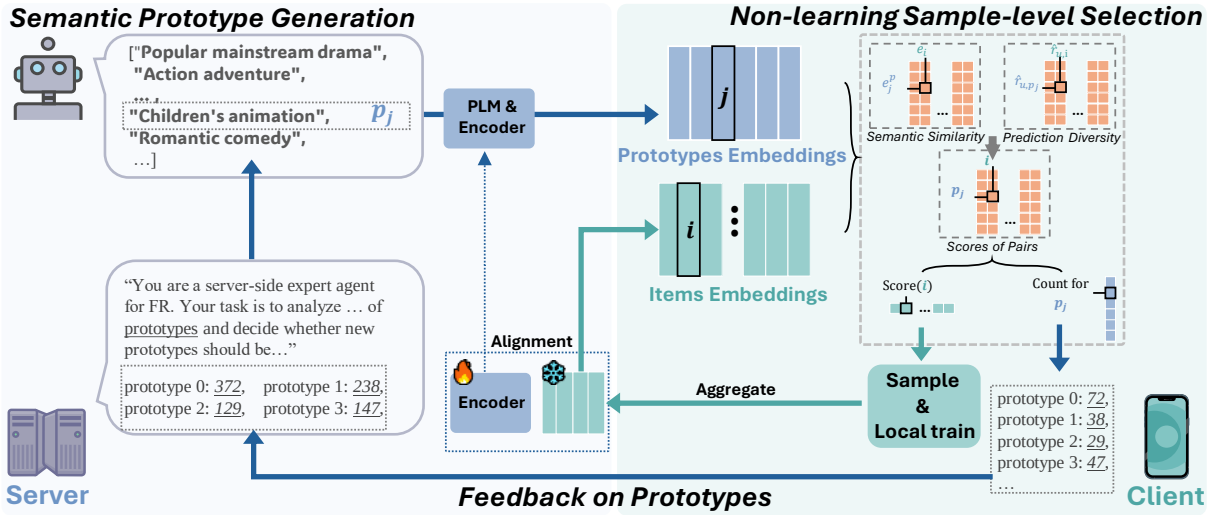


Figure 2: Overall framework of FedSLLM. The server uses an LLM to generate natural language prototypes, which guide clients in selecting samples efficiently. Then clients provide lightweight feedback to update the prototypes.

In the initial federated rounds, when no client feedback is available, the LLM prompt is constructed using a brief summary of the items, including high-level attributes and category statistics.

4.3 Non-learning Sample-level Selection

Given the prototypes from the server, during local training, each selected client u receives the global prototype embeddings E_t^p, the item embedding table E, and the model W_g. For its local interaction set D_u, the client maintains the semantic similarity scores $\mathrm{sim}(i, p_j)$ for all $i \in D_u$ and $p_j \in \mathcal{P}_t$.

To further prioritize informative and challenging samples that are hard to predict, we introduce a diversity-aware criterion based on prediction discrepancy. Let

$$\begin{aligned} \hat{r}_{u,i} &= W_g(u, i), \\ \hat{r}_{u,p_j} &= W_g(u, p_j) \end{aligned} \tag{9}$$

denote the predicted interaction score of client u on item i and prototype p_j by the model, separately. The prediction diversity between item i and prototype p_j is defined as

$$\mathrm{div}(i, p_j) = \left|\hat{r}_{u,i} - \hat{r}_{u,p_j}\right|. \tag{10}$$

Intuitively, a large diversity value indicates that although an item is semantically related to a prototype, the current model fails to generalize its behavior, suggesting that the sample is hard to learn. We combine semantic similarity and prediction diversity to define the sampling score:

$$\mathrm{s}(i, p_j) = \frac{\mathrm{sim}(i, p_j)}{\mathrm{div}(i, p_j) + \epsilon}, \tag{11}$$

where ϵ is a small constant for numerical stability. A larger score indicates that item i is semantically close to prototype p_j while exhibiting a large prediction discrepancy, and is more informative.

To obtain a single sampling score for each item, the client adopts a minimum aggregation strategy over all prototypes:

$$\mathrm{Score}(i) = \min_{j=1,\ldots,N_t} \mathrm{s}(i, p_j). \tag{12}$$

This strategy emphasizes samples that are poorly explained by at least one global semantic prototype, encouraging the model to focus on hard and under-generalized interactions. Finally, each client selects the top-$\mathcal{S}$ fraction of interactions from D_u according to $\mathrm{Score}(i)$ for local model optimization.

4.4 Feedback on Prototypes

Based on the prototype-guided sampling process, for each local interaction sample $i \in D_u$, the client u computes its sampling score with respect to each prototype using Eq. 11.

The closest (i.e., most challenging) prototype for sample i is then identified as

$$j^*(i) = \arg \min_{j \in \{1,\ldots,N_t\}} \mathrm{s}(i, p_j). \tag{13}$$

Based on this assignment, the client constructs a prototype-specific count vector $\mathbf{c}_u \in \mathbb{N}^{n_t}$, where the j-th entry is defined as

$$\mathbf{c}_u[j] = \sum_{i \in D_u} \mathbb{I}\big(j^*(i) = j\big), \tag{14}$$

and $\mathbb{I}(\cdot)$ is the indicator function.

Each entry $\mathbf{c}_u[j]$ therefore reflects how many local samples are insufficiently explained by prototype p_j, providing a coarse-grained but informative signal of prototype difficulty. The server aggregates the count vectors from participating clients to obtain a global prototype difficulty profile:

$$\mathbf{c}_t = \sum_{u \in U_t} \mathbf{c}_u, \tag{15}$$

which summarizes the distribution of hard-to-explain samples across prototypes. Based on this aggregated feedback, the server constructs a concise prompt and queries the LLM to analyze the global prototypes and their count vectors.

4.5 Alignment for Semantic

To support prototype-driven sample selection and semantic guidance, the global aggregated item embeddings in FR must be aligned with the semantic prototype space, as they are learned purely from user-item interactions and lack explicit semantic supervision. To this end, the server periodically updates the AE encoder $\mathcal{E}_{\text{AE}}$ to align semantic and interaction representations.

Specifically, given the semantic representation x_i of item i, the server projects it into the latent space via the encoder $\mathcal{E}_{\text{AE}}$ as in Eq. 3, producing z_i. The encoder is then optimized by aligning z_i with the frozen, aggregated global item embedding $e_i \in E$, which is learned from federated interaction data. The alignment objective is defined as

$$\mathcal{L}_{\mathcal{E}} = \|z_i - e_i\|^2. \tag{16}$$

This alignment strategy enables $\mathcal{E}_{\text{AE}}$ to learn a unified mapping function that bridges semantics and interaction-driven representations. As a result, both semantic prototypes and items can be reliably compared and utilized in a shared latent space.

5 Experiments and Discussions

5.1 Experimental Setup

Datasets. Several public recommendation datasets are used in our experiments: MovieLens-100K, which is denoted as ML100K; MovieLens-1M (Harper and Konstan, 2016) denoted as ML1M; and two Amazon datasets (Ni et al., 2019), Industrial and Software. These datasets offer interaction records along with item attributes (see Table 1). ML100K and ML1M include movie information, while the Amazon datasets provide item descriptions. We adopt the leave-one-out data split method.

Dataset	#Users	#Items	#Interactions	Sparsity
ML100K	943	1,682	100,000	93.70%
ML1M	6,040	3,706	1,000,209	95.53%
Industrial	11,041	5,334	77,071	99.87%
Software	1,826	802	12,805	99.13%

Table 1: Dataset statistics.

Baselines. We compare our method with the following baselines: (1) Random, which randomly samples interactions. (2) Traditional uncertainty-based sampling, including entropy-based sampling (Entropy) (Holub et al., 2008), margin-based sampling (Margin) (Balcan et al., 2007), and least-confidence sampling (Least) (Li and Sethi, 2006). (3) FedSelect (Zhang et al., 2025b), employing an auxiliary meta-model to estimate sample importance. (4) FALE (Tang et al., 2025), which globally computes the heterogeneity-aware sampling scores, represents federated active learning. (5) Our method FedSLLM, which uses LLM-generated semantic prototypes for guiding local sample selection. We evaluate it with two public LLMs, Qwen[1] and LLaMA[2]. For fairness, all baselines adopt the same pre-training strategy as our method.

FR Backbone Models. We adopt four popular FR backbone models: (1) FedMF (Chai et al., 2021), a federated matrix factorization with only user and item embeddings, (2) FedNCF (Perifanis and Efraimidis, 2022), the extension of FedMF with an MLP, (3) FedPerGNN (Wu et al., 2022), a graph-based FR model using only user and item embeddings, and (4) PFedRec (Zhang et al., 2023), a personalized FR model with item embeddings and personalized models without user embeddings.

Evaluation Metrics. We adopt the widely used top-K recommendation evaluation metrics: Hit Ratio (H@K) and Normalized Discounted Cumulative Gain (N@K). Both metrics reflect the ranking quality. Higher values indicate better performance.

Implementation Details. The client sampling ratio $\mathcal{C}$ is set to 10%. The sample selection ratio in clients $\mathcal{S}$ is set to 10%. Each selected client performs 2 local epochs per round, and the total number of global rounds is 200. Our implementation is built on the open-source library FuxiCTR (Zhu et al., 2021). The experiments are conducted with NVIDIA GeForce RTX 4090 GPUs.

[1] https://huggingface.co/Qwen/Qwen2.5-1.5B-Instruct

[2] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

FR Backbone	Sample Method	ML100K			ML1M			Industrial			Software		
		H@5	H@10	H@20	H@5	H@10	H@20	H@5	H@10	H@20	H@5	H@10	H@20
FedNCF	Base	40.40	55.57	73.38	41.06	54.64	77.70	8.16	13.60	23.61	19.50	24.64	34.67
	Random	39.45	51.75	57.90	23.29	41.56	71.80	9.35	14.40	23.68	18.13	23.66	33.30
	Entropy	39.02	51.75	58.32	38.87	45.28	49.70	8.31	13.18	22.56	8.65	13.86	21.69
	Margin	38.49	51.01	58.75	44.34	55.23	59.98	8.73	13.56	23.14	6.46	10.24	19.99
	Least	39.38	45.32	67.41	39.27	46.49	50.43	6.50	11.06	20.69	7.23	12.16	18.78
	FedSelect	30.01	44.86	67.87	46.74	58.36	70.83	16.27	19.88	27.35	16.89	27.24	40.31
	FALE	42.74	58.64	**75.72**	34.67	58.05	71.77	10.42	16.01	26.35	17.80	23.82	34.67
	Ours-Qwen	54.08	61.93	74.66	**51.59**	**60.05**	**74.27**	**23.18**	**32.38**	46.88	**41.68**	**48.69**	**60.73**
	Ours-LLaMa	**60.66**	**65.22**	75.08	48.31	59.44	73.84	20.16	30.41	**47.66**	26.18	36.14	48.25
FedMF	Base	18.24	27.78	41.78	12.48	20.76	33.01	5.57	10.87	20.54	7.78	13.25	23.55
	Random	7.64	14.42	26.51	6.08	11.51	22.50	4.94	10.33	20.57	5.15	10.57	20.97
	Entropy	8.80	15.06	24.60	4.26	8.81	18.84	4.87	9.75	19.88	5.31	11.23	21.80
	Margin	8.38	16.01	26.19	5.96	11.51	21.92	4.68	9.62	20.22	6.85	12.60	21.96
	Least	8.06	16.33	26.09	5.96	11.51	21.92	4.68	9.62	20.22	6.63	12.43	21.69
	FedSelect	28.31	42.84	62.25	25.88	39.00	59.93	6.18	**19.19**	**44.70**	8.38	17.14	31.68
	FALE	6.68	12.62	24.81	6.36	11.54	21.89	5.50	10.62	20.59	5.26	10.68	19.99
	Ours-Qwen	**32.45**	**46.13**	**66.07**	25.58	**39.30**	**61.42**	6.49	18.60	44.53	**8.49**	**19.33**	38.77
	Ours-LLaMa	31.18	43.48	64.26	**26.61**	38.77	61.16	**6.51**	18.86	44.43	6.68	16.87	**40.31**
PFedRec	Base	42.52	58.64	75.82	40.75	53.36	76.89	10.39	16.08	26.46	19.06	24.59	31.82
	Random	40.93	56.42	73.06	39.88	51.42	74.14	8.94	14.96	24.67	17.31	22.23	31.87
	Entropy	38.28	51.75	57.79	40.15	43.64	53.33	8.79	12.92	22.01	17.20	21.85	31.98
	Margin	37.75	48.36	56.42	41.57	54.72	63.77	8.83	13.95	22.61	18.62	23.22	30.78
	Least	37.22	49.73	58.96	42.24	52.28	55.28	9.52	14.27	23.03	17.74	22.51	69.11
	FedSelect	65.96	69.88	80.91	68.03	70.76	83.26	33.91	42.86	57.08	49.62	55.31	34.94
	FALE	25.13	38.07	53.87	66.06	71.27	82.76	38.09	48.61	63.98	17.91	25.90	**70.10**
	Ours-Qwen	68.08	70.41	81.34	73.49	74.55	84.29	**39.42**	**48.75**	**64.42**	**51.70**	**57.34**	67.58
	Ours-LLaMa	**73.49**	**74.13**	**82.82**	**78.81**	**79.93**	**87.91**	35.63	47.70	63.15	49.34	53.72	63.15
FedPerGNN	Base	41.68	57.48	75.61	34.55	58.97	77.04	9.24	14.72	24.59	20.81	27.44	38.77
	Random	38.81	53.45	73.91	27.27	40.84	68.68	8.87	14.69	24.74	17.42	22.62	31.76
	Entropy	38.28	54.29	71.16	29.88	45.00	66.19	8.50	14.51	25.24	18.07	23.77	34.01
	Margin	38.39	54.29	72.85	30.07	44.82	65.98	10.23	16.30	26.93	19.28	26.23	36.31
	Least	35.42	49.84	72.75	26.90	39.07	58.18	9.76	15.32	26.04	19.28	26.23	36.31
	FedSelect	38.49	56.84	67.44	27.55	40.30	66.59	9.65	15.99	26.86	19.33	25.68	35.60
	FALE	39.34	56.95	74.55	29.22	48.59	70.88	9.82	15.61	26.61	18.95	24.81	35.38
	Ours-Qwen	40.08	**57.16**	**74.97**	29.97	43.51	66.14	**11.06**	**17.03**	**27.68**	**19.47**	**30.26**	**37.95**
	Ours-LLaMa	**40.72**	56.31	74.55	**32.10**	**52.47**	**75.48**	10.11	15.88	26.09	18.35	26.94	36.76

Table 2: Performance comparison of different sampling methods in terms of HR (H@K) across various FR backbones and datasets. Best results are highlighted in **bold**, and the second-best results are underlined. "Base" indicates the results obtained using all samples without any sample-level selection.

5.2 Experimental Results

Sample-level selection improves recommendation performance. As the average results in Table 2 on H@K and Table 3 on N@K, applying sample-level selection sometimes leads to better recommendation performance compared to training on all local interactions. For example, on ML100K with FedNCF, the Base model achieves H@20 of 73.38%, while FALE improves it to 75.72% and our method improves to 74.66% and 75.08%. On Industrial-FedNCF, Random, FedSelect, FALE, and our method achieve better performance on all metrics. Even traditional methods can achieve performance comparable to the Base model with a limited sampling ratio ($\mathcal{S} = 10\%$) on the Industrial dataset using PFedRec and FedPerGNN.

Similar improvements can be observed across multiple settings, indicating that avoiding unnecessary training on low-value interactions helps clients focus on more informative samples. These results confirm that sample-level selection is a practical and effective strategy for enhancing FR.

Consistent improvements across backbones and datasets of FedSLLM. As shown in Table 2, our method consistently achieves strong performance across all four FR backbones and three datasets. In particular, it attains the best results on most metrics for FedNCF, PFedRec, and FedPerGNN, and remains competitive on FedMF. For instance, on ML1M with PFedRec, our method improves H@10

FR Backbone	Sample Method	ML100K			ML1M			Industrial			Software		
		N@5	N@10	N@20	N@5	N@10	N@20	N@5	N@10	N@20	N@5	N@10	N@20
FedNCF	Base	28.95	33.88	38.41	29.40	33.71	39.59	5.75	7.48	9.98	16.11	17.75	20.27
	FedSelect	20.73	25.53	31.27	33.27	34.84	38.33	4.33	8.89	13.35	13.47	17.95	21.59
	FALE	31.76	36.93	41.27	22.28	29.79	35.37	7.47	9.26	11.85	13.93	15.86	18.58
	Ours-Qwen	35.08	37.73	40.99	**34.15**	**37.02**	**40.56**	**13.32**	**16.38**	**20.09**	**27.19**	**29.58**	**32.72**
	Ours-LLaMa	**39.78**	**41.36**	**43.91**	32.08	35.07	38.43	11.37	14.69	18.59	13.89	17.28	23.89
FedMF	Base	12.36	15.37	18.90	7.54	10.19	13.27	3.41	5.09	7.51	5.20	6.97	9.54
	FedSelect	15.03	19.87	24.72	14.29	18.85	23.89	3.45	7.59	13.86	5.58	8.38	14.36
	FALE	4.02	5.91	8.95	3.86	5.52	8.10	3.19	4.83	7.33	3.17	4.90	7.22
	Ours-Qwen	**18.05**	**22.60**	**27.59**	14.32	18.63	23.92	3.72	7.58	13.80	**5.71**	**9.15**	**14.66**
	Ours-LLaMa	17.21	21.30	26.48	**14.64**	**18.65**	**24.22**	**3.81**	**7.73**	**14.00**	3.93	7.15	12.90
PFedRec	Base	31.35	36.60	40.95	30.78	34.80	40.75	7.38	9.20	11.79	15.78	17.56	20.06
	FedSelect	44.81	45.44	48.30	40.55	41.50	45.03	18.83	21.88	25.60	30.94	32.89	36.51
	FALE	17.09	21.26	25.25	43.91	44.19	46.28	20.64	24.23	28.21	15.45	18.01	20.29
	Ours-Qwen	43.77	44.58	47.42	51.15	51.51	46.55	**21.69**	**24.89**	**28.95**	**32.04**	**33.98**	**37.33**
	Ours-LLaMa	**50.73**	**50.95**	**53.23**	**56.60**	**56.99**	**59.08**	19.39	23.48	27.46	27.18	28.69	32.35
FedPerGNN	Base	30.56	35.69	40.28	22.28	30.09	34.75	6.71	8.46	10.93	17.26	19.39	22.22
	FedSelect	28.74	30.15	34.61	20.25	24.27	30.85	6.96	8.98	11.69	16.36	18.38	20.87
	FALE	28.92	34.36	38.89	19.95	26.12	32.97	7.01	8.85	11.59	16.33	18.20	20.86
	Ours-Qwen	**29.19**	**34.86**	**39.26**	20.19	24.54	30.26	**7.91**	**9.82**	**12.49**	**17.80**	**19.13**	**22.83**
	Ours-LLaMa	28.71	34.49	39.12	**21.84**	**28.38**	**33.13**	7.81	9.58	12.17	17.20	18.99	21.89

Table 3: Performance comparison in terms of NDCG (N@K). More details are in Appendix Sec B.1.

FR Backbone	Variant	ML100K		ML1M		Industrial		Software	
		H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10
FedNCF	Base	55.57	33.88	54.64	33.71	13.60	7.48	24.64	17.75
	Ours-Qwen	61.93(+11.5)	37.73(+11.3)	60.05(+9.9)	37.02(+9.8)	32.38(+138.0)	16.38(+118.9)	48.69(+97.6)	29.58(+66.6)
	Ours-LLaMa	65.22(+17.4)	41.36(+22.1)	59.44(+8.8)	35.07(+4.0)	30.41(+123.6)	14.69(+96.3)	36.14(+46.7)	17.28(-2.7)
	Ours-Cluster	59.81(+7.6)	35.97(+6.2)	57.07(+4.5)	32.87(-2.5)	30.25(+122.4)	13.76(+83.7)	34.23(+38.9)	15.95(-10.1)
FedMF	Base	27.78	15.37	20.76	10.19	10.87	5.09	13.25	6.97
	Ours-Qwen	46.13(+66.0)	22.60(+47.1)	39.30(+89.3)	18.63(+82.8)	18.60(+71.2)	7.58(+48.9)	19.33(+45.8)	9.15(+31.4)
	Ours-LLaMa	43.48(+56.5)	21.30(+38.6)	38.77(+86.8)	18.65(+83.0)	18.86(+73.5)	7.73(+51.9)	16.87(+27.3)	7.15(+2.7)
	Ours-Cluster	38.92(+40.1)	17.70(+15.1)	36.14(+74.1)	17.38(+70.6)	15.42(+41.9)	6.44(+26.5)	7.83(-40.9)	3.47(-50.3)
PFedRec	Base	58.64	36.60	53.36	34.80	16.08	9.20	24.59	17.56
	Ours-Qwen	70.41(+20.1)	44.58(+21.8)	74.55(+39.7)	51.51(+48.1)	48.75(+203.3)	24.89(+170.5)	57.34(+133.2)	33.98(+93.5)
	Ours-LLaMa	74.13(+26.4)	50.95(+39.2)	79.93(+49.8)	56.99(+63.8)	47.70(+196.7)	23.48(+155.2)	53.72(+118.5)	28.69(+63.4)
	Ours-Cluster	55.99(-4.5)	31.35(-14.3)	65.15(+22.1)	43.44(+24.9)	30.07(+87.0)	13.80(+50.0)	34.23(+39.2)	15.12(-13.9)
FedPerGNN	Base	57.48	35.69	58.97	30.09	14.72	8.46	27.44	19.39
	Ours-Qwen	57.16(-0.6)	34.86(-2.3)	43.51(-26.2)	24.54(-18.4)	17.03(+15.7)	9.82(+16.1)	30.26(+10.3)	19.13(-1.3)
	Ours-LLaMa	56.31(-2.0)	34.49(-3.4)	52.47(-11.0)	28.38(-5.7)	15.88(+7.9)	9.58(+13.3)	26.94(-1.8)	18.99(-2.1)
	Ours-Cluster	56.63(-1.5)	34.74(-2.7)	39.93(-32.3)	24.13(-19.8)	15.22(+3.4)	8.88(+5.00)	22.95(-16.4)	16.91(-12.8)

Table 4: Performance comparison the variants of FedSLLM. Numbers in () indicate relative changes with Base, with improvements in cyan and drops in orange, where intensity scales with the magnitude of change.

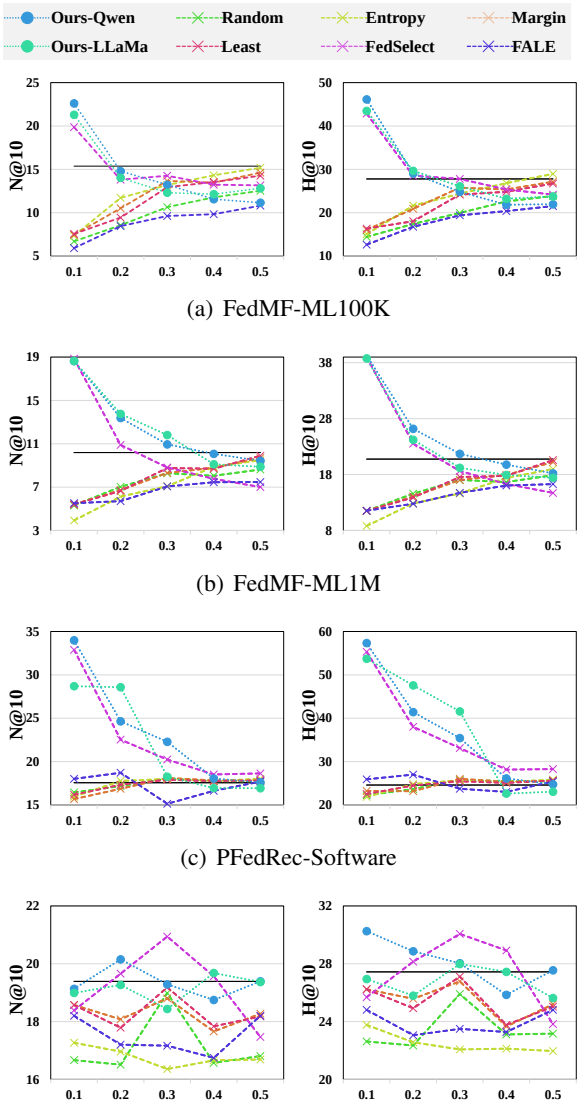
from 53.36% (Base) to 74.55% and 79.93%. While on the Industrial dataset with FedNCF, H@10 increases substantially from 13.60% to 32.38% and 30.41%. Although in some settings our method does not show the best performance, it still shows competitive performance.

These results demonstrate that the proposed prototype-guided sampling strategy is model-agnostic and robust to different data scales and recommendation architectures.

5.3 Ablation Study

To investigate the effect of LLMs, we conduct an ablation study comparing with a variant where prototypes are obtained via clustering item embeddings (*Ours-Cluster*) in Table 4.

While *Ours-Cluster* shows that prototype-based sample selection can yield limited improvements, its effectiveness is clearly constrained. In contrast, LLM-generated semantic prototypes consistently deliver substantial performance gains across datasets and backbones. For instance, in ML100K-PFedRec, FedSLLM with Qwen and LLaMA improves H@10 by 20.1% and 26.4%, respectively, while the clustering-based variant suffers a 4.52% decrease. These results demonstrate that LLM-provided semantic knowledge is crucial for constructing high-quality prototypes.



(a) FedMF-ML100K

(b) FedMF-ML1M

(c) PFedRec-Software

(d) FedPerGNN-Software

Figure 3: Sensitivity study on hyperparameter sample ratio $\mathcal{S}$ in various backbone-dataset settings. The black line shows the base result without sampling.

5.4 Sensitivity Analysis

Robustness across different LLMs. We further investigate the sensitivity of our framework to different LLMs. According to the results in Tables 2, 3 and 4, we observe that LLaMA performs better in ML100K and ML1M datasets overall, while the smaller Qwen model outperforms LLaMA in others. Although different LLMs introduce certain performance fluctuations, our method consistently achieves great improvements over the baselines across various settings. This observation indicates that the effectiveness of FedSLLM is not highly sensitive to the specific LLM, suggesting that the performance gains mainly come from the proposed sample selection design.

Method	Comm. (MB)	Compute. (MFLOPs)	Selection Cost. (ms)
Base	0.2689	0.1645	–
Random	0.2689	0.0164	0.271
FedSelect	1.0130	0.0164	0.313
FALE	0.3324	0.0329	0.747
FedSLLM	0.3330	0.0164	0.384

Table 5: Communication and computation overhead.

Effectiveness at low sampling ratios. Figure 3 illustrates the impact of the sampling ratio $\mathcal{S}$ on recommendation performance. A common observation across most baselines is that their performance generally improves as $\mathcal{S}$ increases. This trend indicates that traditional sampling strategies are limited in distinguishing informative samples from noisy ones. In contrast, our method achieves strong performance, especially at small ratios (e.g., $\mathcal{S} = 0.1$) across various backbones and datasets.

5.5 Discussion on Efficiency

Table 5 compares the communication (model and data size/MB), computation (MFLOPs), and the average sample selection time (ms) across different methods. Our method FedSLLM maintains the small communication size (0.3330 MB) and low computational cost (0.0164 MFLOPs). While the selection time (0.384 ms) is slightly higher than traditional methods represented by Random (0.271 ms) and other state-of-the-art methods such as FedSelect (0.313 ms), it remains modest and practical for resource-constrained clients. These results demonstrate that FedSLLM can provide semantic guidance for sample selection with minimal additional computation and communication overhead, making it suitable for real-world FR scenarios.

6 Conclusion

This paper proposes FedSLLM, an LLM-enhanced prototype-based sample-level selection method for federated recommendation (FR). Specifically, LLMs are leveraged to generate semantic prototypes that capture challenging patterns for training. Based on them, clients can select more valuable samples for local optimization. To further refine the prototypes, clients provide lightweight feedback to the server. Extensive experiments demonstrate that FedSLLM is both effective and efficient. In future work, we will investigate LLM-based sampling in more FR scenarios, such as in cross-domain and multi-domain settings, where abundant source data may not be available.

Limitations

While FedSLLM demonstrates strong empirical performance, it still faces several limitations that are challenging to address. First, although it shows relatively low sensitivity to the choice of LLM, the quality of semantic prototypes still depends on the reasoning and generation capability of the model, which may introduce additional computational overhead in the central server. Moreover, incorporating stronger reasoning or adaptive feedback from LLMs into the federated optimization loop is constrained by privacy, communication, and efficiency requirements, limiting deeper integration of LLM beyond offline prototype generation.

Ethical Considerations

This work does not involve human subjects, sensitive personal data, or any proprietary datasets. All datasets used in this study are publicly available and commonly used in prior research works. We have taken care to ensure that our methods and results do not raise safety, privacy, or fairness concerns.

References

Maria-Florina Balcan, Andrei Z. Broder, and Tong Zhang. 2007. Margin based active learning. In *20th Annual Conference on Learning Theory, COLT 2007*, volume 4539, pages 35–50.

Di Chai, Leye Wang, Kai Chen, and Qiang Yang. 2021. Secure federated matrix factorization. *IEEE Intell. Syst.*, 36:11–20.

Huancheng Chen and Haris Vikalo. 2024. Heterogeneity-guided client sampling: Towards fast and efficient non-iid federated learning. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024*.

Yann Fraboni, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. 2021. Clustered sampling: Low-variance and improved representativity for clients selection in federated learning. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, volume 139, pages 3407–3416.

F. Maxwell Harper and Joseph A. Konstan. 2016. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5:19:1–19:19.

Alex Holub, Pietro Perona, and Michael C. Burl. 2008. Entropy-based active learning for object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2008*, pages 1–8.

Chaejeong Lee, Jeongwhan Choi, Hyowon Wi, Sung-Bae Cho, and Noseong Park. 2025. SCONE: A novel stochastic sampling to generate contrastive views and hard negative samples for recommendation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining, WSDM 2025*, pages 419–428.

Hanzhe Li, Dazhong Shen, Chao Wang, Yuting Liu, and Jingjing Gu. 2025. Can llms enhance fairness in recommendation systems? A data augmentation approach. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2025*, pages 570–580.

Mingkun Li and Ishwar K. Sethi. 2006. Confidence-based active learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(8):1251–1261.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282.

Mahdi Morafah, Saeed Vahidian, Weijia Wang, and Bill Lin. 2023. FLIS: clustered federated learning via inference similarity for non-iid data distribution. *IEEE Open J. Comput. Soc.*, 4:109–120.

Noorain Mukhtiar, Adnan Mahmood, Yipeng Zhou, Jian Yang, Jing Teng, and Quan Z. Sheng. 2025. Federated Learning at the Forefront of Fairness: A Multifaceted Perspective. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2025), Montreal, Canada, August 16-22, 2025*, pages 10603–10611.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874.

Jianmo Ni, Jiacheng Li, and Julian J. McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, pages 188–197.

Zhenhui Pan, Yawen Li, Zeli Guan, Meiyu Liang, Ang Li, Jia Wang, and Feifei Kou. 2025. RFCSC: communication efficient reinforcement federated learning with dynamic client selection and adaptive gradient compression. *Neurocomputing*, 612:128672.

Vasileios Perifanis and Pavlos S. Efraimidis. 2022. Federated neural collaborative filtering. *Knowl. Based Syst.*, 242:108441.

Tao Qi, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. 2023. Fedsampling: A better sampling strategy for federated learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023*, pages 4154–4162.

Wentao Shi, Jiawei Chen, Fuli Feng, Jizhi Zhang, Junkang Wu, Chongming Gao, and Xiangnan He. 2023. On the theories behind hard negative sampling for recommendation. In *Proceedings of the ACM Web Conference 2023, WWW 2023*, pages 812–822.

Duanxiao Song, Guangyuan Shen, Dehong Gao, Libin Yang, Xukai Zhou, Shirui Pan, Wei Lou, and Fang Zhou. 2023. Fast heterogeneous federated learning with hybrid client selection. In *Uncertainty in Artificial Intelligence, UAI 2023*, volume 216, pages 2006–2015.

Ying-Peng Tang, Chao Ren, Xiaoli Tang, Sheng-Jun Huang, Lizhen Cui, and Han Yu. 2025. Efficient heterogeneity-aware federated active data selection. In *Forty-second International Conference on Machine Learning, ICML 2025*.

Zhiliang Tian, Jingyuan Huang, Zejiang He, Zhen Huang, Menglong Lu, Linbo Qiao, Songzhu Mei, Yijie Wang, and Dongsheng Li. 2025. Llm-based rumor detection via influence guided sample selection and game-based perspective analysis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025*, pages 28402–28414.

Zhao Tong, Yimeng Gu, Huidong Liu, Qiang Liu, Shu Wu, Haichao Shi, and Xiao-Yu Zhang. 2025. Generate first, then sample: Enhancing fake news detection with llm-augmented reinforced sampling. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025*, pages 24276–24290.

Paul Voigt and Axel Von dem Bussche. 2017. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10:10–5555.

Jia Wang, Yawen Li, Yingxia Shao, Zhe Xue, Zeli Guan, Ang Li, and Guanhua Ye. 2025a. Reinforcement active client selection for federated heterogeneous graph learning. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence*, pages 21117–21125.

Lei Wang, Songheng Zhang, Yun Wang, Ee-Peng Lim, and Yong Wang. 2023a. Llm4vis: Explainable visualization recommendation using chatgpt. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: EMNLP 2023*, pages 675–692.

Lin Wang, Yongxin Guo, Tao Lin, and Xiaoying Tang. 2023b. DELTA: diverse client sampling for fasting federated learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*.

Shijie Wang, Wenqi Fan, Yue Feng, Shanru Lin, Xinyu Ma, Shuaiqiang Wang, and Dawei Yin. 2025b. Knowledge graph retrieval-augmented generation for llm-based recommendation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025*, pages 27152–27168.

Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Tao Qi, Yongfeng Huang, and Xing Xie. 2022. A federated graph neural network framework for privacy-preserving personalization. *Nature Communications*, 13:3091.

Haoran Xu, Jiaze Li, Wanyi Wu, and Hao Ren. 2025a. Federated learning with sample-level client drift mitigation. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence*, pages 21752–21760.

Haozhe Xu, Xiaohua Wang, Changze Lv, and Xiaoqing Zheng. 2025b. Beyond single labels: Improving conversational recommendation through llm-powered data augmentation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025*, pages 15573–15590.

Liu Yang, Ben Tan, Vincent W. Zheng, Kai Chen, and Qiang Yang. 2020. Federated recommendation systems. In *Federated Learning - Privacy and Incentive*, volume 12500, pages 225–239.

Haochen Yuan, Yang Zhang, Quan Z. Sheng, Lina Yao, Yipeng Zhou, Xiang He, and Zhongjie Wang. 2025a. PKGRec: Personal Knowledge Graph Construction and Mining for Federated Recommendation Enhancement. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM 2025), Seoul, Republic of Korea, November 10-14, 2025*, pages 3973–3982.

Wei Yuan, Chaoqun Yang, Guanhua Ye, Tong Chen, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2025b. FELLAS: enhancing federated sequential recommendation with LLM as external services. *ACM Trans. Inf. Syst.*, 43(6):144:1–144:24.

Chunxu Zhang, Guodong Long, Tianyi Zhou, Peng Yan, Zijian Zhang, Chengqi Zhang, and Bo Yang. 2023. Dual personalization on federated recommendation. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023*, pages 4558–4566.

Sai Qian Zhang, Jieyu Lin, and Qi Zhang. 2022. A multi-agent reinforcement learning approach for efficient client selection in federated learning. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022*, pages 9091–9099.

Xinyu Zhang, Linmei Hu, Luhao Zhang, Wentao Cheng, Yashen Wang, Ge Shi, Chong Feng, and Liqiang Nie. 2025a. Bi-tuning with collaborative information for controllable llm-based sequential recommendation. In *Proceedings of the 63rd Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, pages 19340–19351.

Yan Zhang, Xiaoye Miao, Bin Li, Yangyang Wu, and Yongheng Shang. 2025b. Proxy-validated importance-aware federated sample selection with meta learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.2, KDD 2025*, pages 3855–3866.

Jujia Zhao, Wenjie Wang, Chen Xu, See-Kiong Ng, and Tat-Seng Chua. 2025. A federated framework for llm-based recommendation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2852–2865.

Jieming Zhu, Jinyang Liu, Shuai Yang, Qi Zhang, and Xiuqiang He. 2021. Open benchmarking for click-through rate prediction. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management*, pages 2759–2769.

Chen-Chen Zong, Tong Jin, and Sheng-Jun Huang. 2025. Inconsistency-based federated active learning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025*, pages 7300–7308.

A Insight of LLMs

A.1 LLM Prompts for Prototype Generation and Updating

We employ a server-side large language model (LLM) to generate and update global semantic prototypes used for guiding local sample selection in federated recommendation. The LLM is only accessed by the server and is never involved in client-side training or inference.

Prototype Generation. When initializing the prototype set, the server prompts the LLM to generate a bounded number of high-level semantic prototypes. The prompt specifies that each prototype should represent a distinct item archetype and be expressed as a concise natural-language abstraction rather than specific item names. Key constraints included in the prompt are: (1) prototypes must be semantically independent and non-overlapping, (2) prototypes should reflect common and meaningful item usage patterns, and (3) the output must be strictly formatted as a list of textual prototype descriptions, without explanations or additional metadata (e.g., *"Action adventure", "Romantic comedy", "Sci-fi"*).

Prototype Generation Prompt (Simplified).
`You are a server-side expert for federated recommender systems. Given global item information, generate a compact set of semantic prototypes, where each prototype represents a distinct item archetype. The number of prototypes should be limited, and each prototype should be semantically independent and non-overlapping with others. Use concise, high-level natural language descriptions rather than specific item names. Return the prototypes as a list of textual descriptions (e.g., "Action adventure", "Romantic comedy", "Sci-fi").`

Prototype Updating. For prototype updating, the server provides the LLM with the current prototype set and their aggregated usage frequencies from the previous training round. The prompt instructs the LLM to: (1) retain prototypes with stable and high usage, (2) merge prototypes that exhibit low usage and semantic overlap, (3) split frequently used but overly coarse prototypes into finer-grained ones, and (4) introduce new prototypes only when important semantic patterns are under-represented. The updated prototype set is returned in the same constrained list format as in prototype generation.

These prompt designs allow the server to maintain a compact, adaptive, and semantically meaningful prototype space while ensuring stable integration with downstream federated training.

Prototype Updating Prompt (Simplified).
`Given the current set of global semantic prototypes and their usage statistics from the previous training round {prototype: count}, update the prototypes to better reflect their effectiveness. You may keep, merge, split, or generate prototypes based on their usage frequency and representational quality. The updated prototypes should remain concise, semantically distinct, and expressed as high-level natural language descriptions. Return the updated prototypes as a list of textual descriptions.`

B Experiments and Discussions

B.1 Additional Experimental Results

We report additional results on N@K in Table 6. Higher N@K indicates better ranking quality, especially for top-ranked recommendations.

From the results, we observe that FedSLLM consistently achieves the best performance in most cases. Although the two variants of FedSLLM do

FR Backbone	Sample Method	ML100K			ML1M			Industrial			Software		
		N@5	N@10	N@20	N@5	N@10	N@20	N@5	N@10	N@20	N@5	N@10	N@20
FedNCF	Base	28.95	33.88	38.41	29.40	33.71	39.59	5.75	7.48	9.98	16.11	17.75	20.27
	Random	26.54	30.54	32.12	15.38	21.17	28.83	6.83	8.45	10.77	15.43	17.22	19.64
	Entropy	27.33	31.47	33.14	28.73	30.81	31.93	6.02	7.58	9.92	5.19	6.86	8.83
	Margin	26.72	30.79	32.74	32.08	35.64	36.85	6.54	8.07	10.47	4.22	5.42	7.83
	Least	26.25	30.30	31.39	28.04	30.38	31.40	3.99	5.45	7.84	4.30	5.88	7.52
	FedSelect	20.73	25.53	31.27	33.27	34.84	38.33	4.33	8.89	13.35	13.47	17.95	21.59
	FALE	31.76	36.93	41.27	22.28	29.79	35.37	7.47	9.26	11.85	13.93	15.86	18.58
	Ours-Qwen	35.08	37.73	40.99	**34.15**	**37.02**	**40.56**	**13.32**	**16.38**	**20.09**	**27.19**	**29.58**	**32.72**
	Ours-LLaMa	**39.78**	**41.36**	**43.91**	32.08	35.07	38.43	11.37	14.69	18.59	13.89	17.28	23.89
FedMF	Base	12.36	15.37	18.90	7.54	10.19	13.27	3.41	5.09	7.51	5.20	6.97	9.54
	Random	4.53	6.69	9.72	3.57	5.31	8.06	2.83	4.56	7.12	3.16	4.91	7.50
	Entropy	5.33	7.35	9.75	2.47	3.92	6.42	2.86	4.42	6.94	3.19	5.07	7.71
	Margin	5.01	7.45	10.00	3.64	5.41	8.01	2.75	4.33	6.97	4.15	5.99	8.34
	Least	4.92	7.56	10.00	3.64	5.41	8.01	2.75	4.33	6.97	4.05	5.91	8.22
	FedSelect	15.03	19.87	24.72	14.29	18.85	23.89	3.45	7.59	13.86	5.58	8.38	14.36
	FALE	4.02	5.91	8.95	3.86	5.52	8.10	3.19	4.83	7.33	3.17	4.90	7.22
	Ours-Qwen	**18.05**	**22.60**	**27.59**	14.32	18.63	23.92	3.72	7.58	13.80	**5.71**	**9.15**	**14.66**
	Ours-LLaMa	17.21	21.30	26.48	**14.64**	**18.65**	**24.22**	**3.81**	**7.73**	**14.00**	3.93	7.15	12.90
PFedRec	Base	31.35	36.60	40.95	30.78	34.80	40.75	7.38	9.20	11.79	15.78	17.56	20.06
	Random	30.51	35.48	39.69	24.10	27.85	33.53	6.44	8.35	10.78	14.87	16.44	18.82
	Entropy	25.13	29.55	31.11	27.97	29.12	31.46	6.54	7.86	10.13	14.47	15.95	18.46
	Margin	25.88	29.32	31.38	27.61	31.82	34.17	6.49	8.12	10.29	14.21	15.67	17.86
	Least	25.55	29.70	32.01	30.45	33.71	34.49	7.04	8.56	10.75	14.67	16.18	18.23
	FedSelect	44.81	45.44	48.30	40.55	41.50	45.03	18.83	21.88	25.60	30.94	32.89	36.51
	FALE	17.09	21.26	25.25	43.91	44.19	46.28	20.64	24.23	28.21	15.45	18.01	20.29
	Ours-Qwen	43.77	44.58	47.42	51.15	51.51	46.55	**21.69**	**24.89**	**28.95**	**32.04**	**33.98**	**37.33**
	Ours-LLaMa	**50.73**	**50.95**	**53.23**	**56.60**	**56.99**	**59.08**	19.39	23.48	27.46	27.18	28.69	32.35
FedPerGNN	Base	30.56	35.69	40.28	22.28	30.09	34.75	6.71	8.46	10.93	17.26	19.39	22.22
	Random	28.04	34.69	38.97	20.15	24.23	31.26	6.09	7.97	10.47	15.01	16.67	18.97
	Entropy	27.78	32.69	37.17	19.73	24.59	29.97	6.14	8.05	10.73	15.44	17.27	19.84
	Margin	28.01	32.93	37.64	19.87	24.61	29.98	7.39	9.33	11.99	16.36	18.58	21.08
	Least	25.49	33.13	37.83	19.14	23.03	27.84	6.97	8.75	11.43	16.36	18.58	21.08
	FedSelect	28.74	30.15	34.61	20.25	24.27	30.85	6.96	8.98	11.69	16.36	18.38	20.87
	FALE	28.92	34.36	38.89	19.95	26.12	32.97	7.01	8.85	11.59	16.33	18.20	20.86
	Ours-Qwen	**29.19**	**34.86**	**39.26**	20.19	24.54	30.26	**7.91**	**9.82**	**12.49**	**17.80**	**19.13**	**22.83**
	Ours-LLaMa	28.71	34.49	39.12	**21.84**	**28.38**	**33.13**	7.81	9.58	12.17	17.20	18.99	21.89

Table 6: Performance comparison of different sampling methods in terms of NDCG (N@K) across various FR backbones and datasets. Best results are highlighted in **bold**, and the second-best results are underlined. "Base" indicates the results obtained using all samples without any sample-level selection.

not attain the best or second-best results in all settings, they still deliver competitive performance in general.

For example, in the Industrial-PFedRec setting, Ours-Qwen achieves the best performance with an N@10 of 24.89%, while FALE obtains the second-best result with an N@10 of 24.23%. Ours-LLaMa also demonstrates competitive performance, achieving an N@10 of 23.48%, which represents a significant performance improvement over the traditional methods.

Moreover, we observe that although our method and several baselines, such as FedSelect and FALE, achieve improvements in most cases, the gains can depend on the underlying FR backbone. For example, on FedPerGNN, the improvement is relatively limited, whereas on PFedRec, the improvement is substantial. Ours-LLaMa achieves a 63.8% relative increase in N@10 compared with the Base model.

B.2 Implementation Details

We adopt leave-one-out strategy while processing datasets, training the models, and evaluating the results.

For all federated recommendation (FR) backbones, the embedding dimension k for both users and items is set to 32. In FedNCF, the MLP architecture is $[64, 128, 64]$ with ReLU activation and a

dropout rate of 0.5. For FedPerGNN, the number of message-passing steps is set to 1. In PFedRec, the MLP architecture is $[32, 64, 32]$ with ReLU activation and a dropout rate of 0.5.

We employ the Adam optimizer for all experiments, with the learning rate set in $\{1e-3, 5e-3, 1e-2\}$ and weight decay of $\{1e-5, 1e-6, 1e-7, 1e-8\}$. The best result across various configurations is reported.

We leverage the pre-trained language model Sentence-T5 (Ni et al., 2022) to encode textual item attributes (e.g., titles and descriptions), which are then fused into the initial embeddings with dimension $k_p = 768$.

For all methods, item embeddings are initialized using an autoencoder trained on side information. The encoder $\mathcal{E}_{AE}$ consists of fully connected layers $[768, 512, 256, 128, 32]$ with ReLU activation and the decoder is reversed. Pre-training is conducted for a number of rounds chosen from $\{10^3, 10^4, 10^5, 10^6\}$, with a learning rate selected from $\{1e-3, 1e-4\}$.