Appendix of Personalized Federated Collaborative Filtering: A Variational AutoEncoder Approach

Anonymous submission

Appendix

Method Discussion

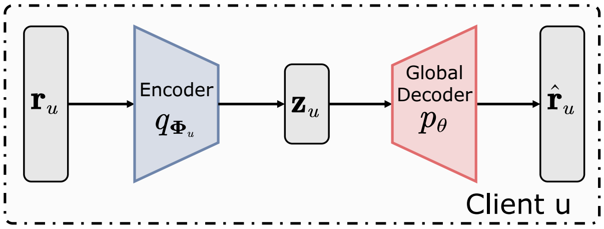

Figure A1: The abstract architecture of FedDAE on the u-th client. When the encoding part is abstracted into a single encoder, the model structure of FedDAE on the u-th client is similar to that of Mult-VAE.

In the Methods section, we introduced that FedDAE constructs a VAE with dual encoders on each client and combines the outputs of these encoders weighted according to the user's data on the client. Therefore, as shown in Fig. A1, let us first define the encoding part of FedDAE as an abstract encoder q, driven by three variables: φ, φ_u, and ψ, collectively represented as $\mathbf{\Phi}_u$. Therefore, we have:

$$\mathbf{z}_u \sim q_{\mathbf{\Phi}_u}(\mathbf{z}_u|\mathbf{r}_u), \tag{A1}$$

which provides the theoretical basis for the construction of Eq. (5).

Now, revisiting Figure 2, we see that the latent variable $\mathbf{z}_u$ on each client is determined by the outputs of both the global encoder q_φ and the local encoder q_{φ_u}. Let $\mathbf{z}_g^{(u)} \sim q_\varphi(\mathbf{z}_g^{(u)}|\mathbf{r}_u)$ represent the output of the global encoder q_φ and $\mathbf{z}_l^{(u)} \sim q_{\varphi_u}(\mathbf{z}^{(u)_l}|\mathbf{r}_u)$ represent the output of the local encoder q_{φ_u}. Therefore, we can express $\mathbf{z}^{(u)}$ as:

$$\mathbf{z}^{(u)} = w_{u1} \cdot \mathbf{z}_g^{(u)} + w_{u2} \cdot \mathbf{z}_l^{(u)} \tag{A2}$$

where the weights w_{u1} and w_{u2} are generated by the gating network h_{ψ_u} based on the user interaction data $\mathbf{r}_u$ on each client. Therefore, by citing Lemma 1, we have:

$$\begin{aligned}\bar{\mu}(\mathbf{r}_u) &= \omega_{u1} \cdot \mu_\varphi(\mathbf{r}_u) + \omega_{u2} \cdot \mu_{\varphi_u}(\mathbf{r}_u),\\ \bar{\sigma}^2(\mathbf{r}_u) &= \omega_{u1}^2 \cdot \sigma_\varphi^2(\mathbf{r}_u) + \omega_{u2}^2 \cdot \sigma_{\varphi_u}^2(\mathbf{r}_u).\end{aligned} \tag{A3}$$

Thus, we have derived Eq. (4) of the main paper.

Algorithm Optmization

In Algorithm 1, φ and θ are updated using the accumulated gradients $\nabla_\varphi^{(u)} q$ and $\nabla_\theta^{(u)} p$ uploaded by each client through the gradient descent algorithm. In this section, we will briefly discuss the rationale behind this update method.

From lines 3 and 4 of the $\mathrm{ClientUpdate}$ function in Alg. 1, it is evident that φ and θ are updated locally on each client. Taking θ as an example, let θ^t be the value of θ after the t-th local iteration. The update rule can be written as:

$$\begin{aligned}\theta^{(t)} &= \theta^{(t-1)} - \eta \cdot \nabla_\theta f^{(t)}\\ &= \theta^{(t-2)} - \eta \cdot \sum_{i=t-1}^{t} \nabla_\theta \mathcal{L}_\beta\\ &= \cdots\\ &= \theta^{(0)} - \eta \cdot \sum_{i=1}^{t} \nabla_\theta \mathcal{L}_\beta,\end{aligned} \tag{A4}$$

where $\theta^{(0)}$ is initialized by the client using the received θ for the current communication round. Since the value of η remains constant during local updates, line 6 of the $\mathrm{ClientUpdate}$ function in Alg. 1 can be expressed as:

$$\nabla_\theta = \sum_{i=1}^{t} \nabla_\theta \mathcal{L}_\beta = \frac{1}{\eta}(\theta^{(0)} - \theta^{(t)}). \tag{A5}$$

Therefore, the accumulated gradient ∇_θ can be used to update θ in the Global Procedure. Similarly, the accumulated gradient ∇_φ can be used to update φ in the Global Procedure.

Convergence Analysis

In this section, We draw on the convergence analysis from work (Tan et al. 2022; Yi et al. 2024) to discuss the convergence of FedDAE. For notational simplicity, we use $\mathbf{\Theta}_u$ to represent $\{\varphi, \varphi_u, \psi_u, \theta\}$, use $\mathcal{L}_u^{(t)} = \mathcal{L}_\beta(\mathbf{r}_u; \mathbf{\Theta}_u^{(t)})$ for the u-th user at the t-th communication round, and let $g_u^{(t)} = \nabla \mathcal{L}_\beta(\mathcal{B}_u^{(t)}; \mathbf{\Theta}_u^{(t)})$, where $\mathcal{B}_u^{(t)}$ is a batch of local data $\mathbf{r}_u$ at the t-th communication round. We denote $e \in \{0, 1/2, 1, 2, \cdots, E\}$ as the local iteration, and $tE + e$ is the e-th local update in the $(t+1)$-th communication round. $e = 0$ denotes that the time step when the client receives the global shared parameters φ and θ.

Assumption 1 (Lipschitz Smoothness) *Gradients of the u-th user's local model are L_1-Lipschitz continuous, i.e.,*

$$\|\nabla\mathcal{L}_u^{(t_1)} - \nabla\mathcal{L}_u^{(t_2)}\|_2 \leq L_1\|\mathbf{\Theta}_u^{(t_1)} - \mathbf{\Theta}_u^{(t_2)}\|, \\ \forall t_1, t_2 > 0, u \in \{1, 2, \cdots, n\}. \tag{A6}$$

The above formula can be further derived as the following quadratic bound:

$$\mathcal{L}_u^{(t_1)} - \mathcal{L}_u^{(t_2)} \leq \langle\nabla\mathcal{L}_u^{(t_2)}, (\mathbf{\Theta}_u^{(t_1)} - \mathbf{\Theta}_u^{(t_2)})\rangle + \tfrac{L_1}{2}\|\mathbf{\Theta}_u^{(t_1)} - \mathbf{\Theta}_u^{(t_2)}\|, \\ \forall t_1, t_2 > 0, u \in \{1, 2, \cdots, n\}. \tag{A7}$$

Assumption 2 (Unbiased Gradient and Bounded Variance) *The stochastic gradient $g_u^{(t)}$ is unbiased, i.e.,*

$$\mathbb{E}_{\mathcal{B}_u^{(t)} \subseteq \mathbf{r}_u}[g_u^{(t)}] = \nabla\mathcal{L}_u^{(t)}, \forall u \in \{1, 2, \cdots, n\}, \tag{A8}$$

and the variance is bounded by:

$$\mathbb{E}[\|g_u^{(t)} - \nabla\mathcal{L}_u^{(t)}\|_2^2] \leq \tau^2. \tag{A9}$$

Assumption 3 (Bounded Parameter Variation) *The parameter variation of the global components $\upsilon = \{\varphi, \theta\}$ before and after aggregation is bounded as:*

$$\|\upsilon^{(t)} - \upsilon_u^{(t)}\|_2^2 \leq \delta^2, \tag{A10}$$

where $\upsilon_u^{(t)}$ is updated on the u-th client at the t-th communication round.

Based on the above assumptions, we have the following Lemmas according to the work (Tan et al. 2022; Yi et al. 2024).

Lemma 1 (Local Model Training) *When all the above assumptions hold, for an arbitrary client's model in the $(t+1)$-th communication round, we have:*

$$\mathbb{E}[\mathcal{L}_u^{(t+1)}] \leq \mathcal{L}_u^{(tE+0)} + (\tfrac{L_1\eta^2}{2} - \eta)\textstyle\sum_{e=0}^{E}\|\nabla\mathcal{L}_u^{(tE+e)}\|_2^2 + \tfrac{L_1E\eta^2\tau^2}{2}. \tag{A11}$$

Lemma 2 (Server Aggregation) *When Assumption 2 and 3 hold, after the $(t+1)$-th communcation round, the loss of any client before and after aggregating the shared parameter υ at the server is bounded by:*

$$\mathbb{E}[\mathcal{L}_u^{((t+1)E+0)}] \leq \mathbb{E}[\mathcal{L}_u^{(tE+1)}] + \eta\delta^2. \tag{A12}$$

For the detailed proof of these lemmas, please refer to the work (Tan et al. 2022; Yi et al. 2024). When FedDAE meets the above assumptions, we can proceed with the following discussions based on the lemmas 1 and 2.

Discussion 1 (One-round deviation) *Based on Lemma 1 and 2, for any client, after the stages of local training, server aggregation and receiving the new global shared parameters, we have:*

$$\mathbb{E}[\mathcal{L}_u^{((t+1)E+0)}] \leq \\ \mathcal{L}_u^{(tE+0)} + (\frac{L_1\eta^2}{2} - \eta)\sum_{e=0}^{E}\|\nabla\mathcal{L}_u^{(tE+e)}\|_2^2 + \frac{L_1E\eta^2\tau^2}{2} + \eta\delta^2. \tag{A13}$$

Proof 1 *By substituting the right-hand side of the inequality in Lemma 2 with Lemma 1, we can directly obtain Eq. A13.*

Discussion 2 (Non-convex Convergence rate of FedDAE) *Based on the Theorem 1, for any client u and a constant $\rho > 0$, we have the following:*

$$\tfrac{1}{T}\textstyle\sum_{t=0}^{T-1}\sum_{e=0}^{E-1}\|\nabla\mathcal{L}_u^{(tE+e)}\|_2^2 \leq \\ \frac{\frac{1}{T}\sum_{t=0}^{T-1}[\mathcal{L}_u^{(tE+0)} - \mathbb{E}[\mathcal{L}_u^{((t+1)E+0)}]] + \frac{L_1E\eta^2\tau^2}{2} + \eta\tau^2}{\eta - \frac{L_1\eta^2}{2}} < \rho, \\ \mathrm{s.\,t.}\, \eta < \tfrac{2(\rho-\delta^2)}{L_1(\rho+E\delta^2)}. \tag{A14}$$

From Theorem 2, it can be seen that any client model of FedDAE can converge at a non-convex rate of $O(\frac{1}{T})$.

Proof 2 *First, by bringing term $\sum_{e=0}^{E}\|\nabla\mathcal{L}_u^{(tE+e)}\|_2^2$ from Eq. A13 to the left side of the inequality, we obtain:*

$$\sum_{e=0}^{E-1}\|\nabla\mathcal{L}_u^{(tE+e)}\|_2^2 \leq \frac{\mathcal{L}_u^{(tE+0)} - \mathbb{E}[\mathcal{L}_u^{((t+1)E+0)}] + \frac{L_1E\eta^2\tau^2}{2} + \eta\tau^2}{\eta - \frac{L_1\eta^2}{2}}. \tag{A15}$$

Let $\mathcal{L}_u^*$ be the optimal objective of the u-th client, we have:

$$\sum_{t=0}^{T-1}[\mathcal{L}_u^{(tE+0)} - \mathbb{E}[\mathcal{L}_u^{((t+1)E+0)}]] \leq \mathcal{L}_u^{(0)} - \mathcal{L}_u^*. \tag{A16}$$

Thus, we can get:

$$\tfrac{1}{T}\textstyle\sum_{t=0}^{T=1}\sum_{e=0}^{E-1}\|\nabla\mathcal{L}_u^{(tE+e)}\|_2^2 \leq \frac{\frac{1}{T}(\mathcal{L}_u^{(0)} - \mathcal{L}_u^*) + \frac{L_1E\eta^2\tau^2}{2} + \eta\tau^2}{\eta - \frac{L_1\eta^2}{2}}. \tag{A17}$$

Assume that the above equation converges to a constant $\rho > 0$, the inequality becomes:

$$\tfrac{1}{T}\textstyle\sum_{t=0}^{T=1}\sum_{e=0}^{E-1}\|\nabla\mathcal{L}_u^{(tE+e)}\|_2^2 \leq \frac{\frac{1}{T}(\mathcal{L}_u^{(0)} - \mathcal{L}_u^*) + \frac{L_1E\eta^2\tau^2}{2} + \eta\tau^2}{\eta - \frac{L_1\eta^2}{2}} < \rho. \tag{A18}$$

Then,

$$T > \frac{\mathcal{L}_u^{(0)} - \mathcal{L}_u^*}{\rho(\eta - \frac{L_1\eta^2}{2}) - \frac{L_1E\eta^2\tau^2}{2} - \eta\tau^2}. \tag{A19}$$

Since $T > 0$ and $\mathcal{L}_u^{(0)} - \mathcal{L}_u^* > 0$, we can know that:

$$\rho(\eta - \frac{L_1\eta^2}{2}) - \frac{L_1E\eta^2\tau^2}{2} - \eta\tau^2 > 0. \tag{A20}$$

By solving the above inequality, we finally get:

$$\eta < \frac{2(\rho - \delta^2)}{L_1(\rho + E\delta^2)}. \tag{A21}$$

Since ρ, L_1, τ^2 and δ^2 are all constants that greater than 0, η must have solutions. Thus, when the learning rate η satisfies the above condition, and client's local model can converge. We can easily derive that the non-convex convergence rate of FedDAE is $O(\frac{1}{T})$.

More Experiment Details

Experimental Setting

Table 1 in the main paper provides the statistical details of the datasets employed in this study includes the following:

#Ratings represents the number of observed ratings. #Users denotes the number of users. #Items indicates the number of items. Sparsity is the percentage of #Ratings out of the total possible ratings.

All the datasets used in the experiments are publicly available: MovieLens-100K (ML-100K)[1], MovieLens-1M (ML-1M)[1], Amazon-Instant-Video (Video)[2], and QB-article[3]. All methods were implemented using PyTorch (Paszke et al. 2019), and experiments were conducted on a machine equipped with a 2.5GHz 14-Core Intel Core i9-12900H processor, a RTX 3070 Ti Laptop GPU, and 64GB of memory.

More Ablation Study

By varying different components of the FedDAE model, we aim to investigate the impact of each component on the model's performance in this section: (1) **FedDAE**$_{global}$: Aggregates the gradients of all components on the server for global aggregation; (2) **FedDAE**$_{local}$: Keeps all component information local without sharing it with the server;

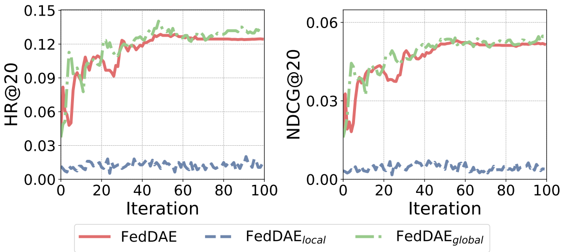

Figure A2: Comparison of the performance of FedDAE with FedDAE$_{local}$ and FedDAE$_{global}$ on the ML-100K dataset.

Ablation Study on Locality and Sharing. Fig. A2 shows the performance of FedDAE and its two variants, FedDAE$_{local}$ and FedDAE$_{global}$, on the ML-100K dataset in terms of HR@20 and NDCG@20 as a function of iterations. As illustrated in the figure, FedDAE achieves the best performance on both HR@20 and NDCG@20 metrics, indicating that its strategy of combining global and local encoders with a gating network is effective. FedDAE$_{local}$ performs poorly due to learning only from local data, while FedDAE$_{global}$, although improved, still suffers from instability and lower effectiveness due to the lack of personalized information for users.

Limitation Discussion

The parameters learned by the two encoders in FedDAE are related to the number of items m in the item set, which may require substantial storage space in practical applications. However, the generative capabilities of VAE might offer a potential solution, such as modeling the feature space of items to generate new ones. Additionally, the FedDAE architecture includes a global encoder, a local encoder, and a gating network, making the model relatively complex. As the number of clients increases, the overall time complexity and space complexity will also significantly increase.

Although FedDAE performs well on datasets with lower sparsity and moderate size (such as ML-100K and ML-1M) as shown in Figure 3, its performance on highly sparse datasets (such as Video and QB-article) is less satisfactory. Therefore, additional strategies may be required to handle such highly sparse datasets.

References

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Tan, Y.; Long, G.; Liu, L.; Zhou, T.; Lu, Q.; Jiang, J.; and Zhang, C. 2022. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 8432–8440.

Yi, L.; Yu, H.; Ren, C.; Zhang, H.; Wang, G.; Liu, X.; and Li, X. 2024. FedMoE: Data-Level Personalization with Mixture of Experts for Model-Heterogeneous Personalized Federated Learning. *arXiv preprint arXiv:2402.01350*.

[1] https://grouplens.org/datasets/movielens/

[2] http://jmcauley.ucsd.edu/data/amazon/

[3] https://github.com/yuangh-x/2022-NIPS-Tenrec