Structured Cooperative Learning with Graphical Model Priors

Shuangtong Li [1] **Tianyi Zhou** [2] **Xinmei Tian** [1 3] **Dacheng Tao** [4]

Abstract

We study how to train personalized models for different tasks on decentralized devices with limited local data. We propose "**S**tructured **Coo**perative **L**earning (**SCooL**)", in which a cooperation graph across devices is generated by a graphical model prior to automatically coordinate mutual learning between devices. By choosing graphical models enforcing different structures, we can derive a rich class of existing and novel decentralized learning algorithms via variational inference. In particular, we show three instantiations of SCooL that adopt Dirac distribution, stochastic block model (SBM), and attention as the prior generating cooperation graphs. These EM-type algorithms alternate between updating the cooperation graph and cooperative learning of local models. They can automatically capture the cross-task correlations among devices by only monitoring their model updating in order to optimize the cooperation graph. We evaluate SCooL and compare it with existing decentralized learning methods on an extensive set of benchmarks, on which SCooL always achieves the highest accuracy of personalized models and significantly outperforms other baselines on communication efficiency. Our code is available at https://github.com/ShuangtongLi/SCooL.

1. Introduction

Decentralized learning of personalized models (**DLPM**) is an emerging problem in a broad range of applications, in which multiple clients target different yet relevant tasks but no central server is available to coordinate or align their learning. A practical challenge is that each single client may not have sufficient data to train a model for its own task and thus has to cooperate with others by sharing knowledge, i.e., through cooperative learning. However, it is usually difficult for a client in the decentralized learning setting to decide when to cooperate with which clients in order to achieve the greatest improvement on its own task, especially when the personal tasks and local data cannot be shared across clients. Moreover, frequently communicating with all other clients is usually inefficient or infeasible. Hence, it is critical to find a sparse cooperation graph only relating clients whose cooperative learning is able to bring critical improvement to their personalization performance. Since the local models are kept being updated, it is also necessary to accordingly adjust the graph to be adaptive to such changes in the training process.

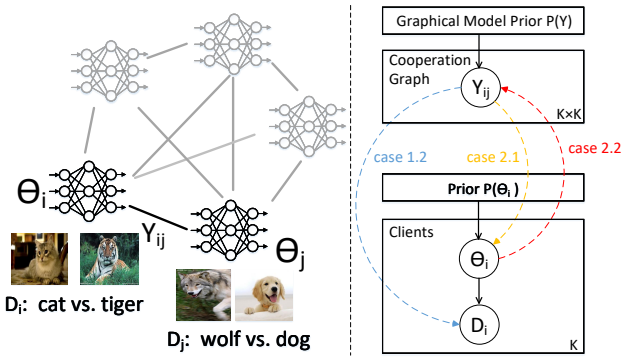


Figure 1. SCooL framework. **Left**: *SCooL* optimizes a $K \times K$ cooperation graph Y together with K personalized models $\theta_{1:K}$ for K tasks, which transfer knowledge across the K clients via decentralized learning. **Right**: *SCooL*'s probabilistic model. Y is generated from a graphical model prior while θ_i or data D_i per client is generated based on Y. We discuss configurations (cases in Section 3.2) of the probabilistic model and derive EM-type algorithms (Section 3.3) that alternately updates Y and $\theta_{1:K}$.

Structural learning of a cooperation graph on the fly with decentralized learning of local models is an open challenge and can be prone to high variance caused by client heterogeneity and local data deficiency. Inspired Bayesian methods and their priors, we propose "**S**tructured **Coo**perative **L**earning (**SCooL**)". SCooL applies a probabilistic graphical model (PGM) as a structured prior enforcing certain structures such as clusters when generating the cooperation graph. By combining such a graphical model prior with the expressive power of neural networks on learning local tasks, we are able to develop a general framework for DLPM, from which

[1]University of Science and Technology of China [2]University of Maryland, College Park [3]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center [4]The University of Sydney. Correspondence to: Tianyi Zhou <tianyi@umd.edu>, Xinmei Tian <xinmei@ustc.edu.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

we can derive a rich class of novel algorithms associated with different structured priors. In particular, we propose a probabilistic model to generate the cooperation graph and local models (Fig. 1 Right). Variational inference on this probabilistic model produces an approximate Maximum-A-Posteriori (MAP) estimation, which leads to an EM-type algorithm that alternately updates the cooperation graph and local models (Fig. 1 Left).

We discuss several designs or configurations of the key components in the generative model and a general variational inference framework to derive EM algorithms for the model. For instance, we apply three different graphical model priors to generate the cooperation graph in the model and follow SCooL framework to derive three decentralized learning algorithms. While the Dirac Delta prior leads to an existing algorithm, i.e., D-PSGD (Lian et al., 2017), the other two priors, i.e., stochastic block model (SBM) and attention, lead to two novel algorithms (*SCooL-SBM* and *SCooL-Attention*) that assume different structures and correlation among the local tasks. These two structural priors accelerate the convergence to a sparse cooperation graph (Fig. 4-5), which can accurately identify the relevant tasks/clients and significantly save the communication cost (Fig. 7).

In experiments on several decentralized learning benchmarks created from three datasets using two different schemes to draw non-IID tasks, SCooL outperforms SOTA decentralized and federated learning approaches on personalization performance (Table 2) and computational/communication efficiency (Fig. 3). We further investigate the capability of SCooL on recovering the cooperation graph pre-defined to draw non-IID tasks. The results explain how SCooL captures the task correlations to coordinate cooperation among relevant tasks and improve their own personalization performance.

2. Related work

Federated learning (FL) (McMahan et al., 2017) Both empirical (Hsieh et al., 2020) and theoretical (Karimireddy et al., 2020) studies find that the performance of FL degrades in non-IID settings when the data distributions (e.g., tasks) over devices are heterogeneous. Several strategies have been studied to address the non-IID challenge: modifying the model aggregation (Lin et al., 2020; Fraboni et al., 2021; Chen & Chao, 2021; Wang et al., 2020; Balakrishnan et al., 2022), regularizing the local objectives with proximal terms (Acar et al., 2021; Li et al., 2018). or alleviating catastrophic forgetting in local training (Xu et al., 2022). These methods focus on improving the global model training to be more robust to non-IID distributions they can be sub-optimal for training personalized models for local tasks. Recent works study to improve the personalization performance in non-IID FL via: (1) trading-off between the global model and local personalization (Li et al., 2021a; T. Dinh et al., 2020); (2) clustering of local clients (Sattler et al., 2020; Ghosh et al., 2020; Xie et al., 2021; Long et al., 2022); (3) personalizing some layers of local models (Li et al., 2021b; Liang et al., 2020; Collins et al., 2021; Oh et al., 2022; Zhang et al., 2023); (4) knowledge distillation (Zhu et al., 2021; Afonin & Karimireddy, 2022); (5) training the global model as an initialization (Fallah et al., 2020) or a generator (Shamsian et al., 2021) of local models; (6) using personalized prototypes (Tan et al., 2022a;b); (7) masking local updates (Dai et al., 2022); or (8) learning the collaboration graph (Chen et al., 2022). Most of them focus on adjusting the interactions between the global model and local personalized models. In contrast to DLPM (our problem), FL assumes a global server so direct communication among personalized models is not fully explored. Although clustering structures have be studied for FL, structure priors of cross-client cooperation graphs has not been thoroughly discussed.

Decentralized learning (DL) Earlier works in this field combine the gossip-averaging (Blot et al., 2016) with SGD. Under topology assumptions such as doubly stochastic mixing-weights (Jiang et al., 2017), all local models can be proved to converge to a "consensus model" (Lian et al., 2017) after iterating peer-to-peer communication. Although they show promising performance in the IID setting, (Hsieh et al., 2020) points out that they suffer from severe performance degeneration in non-IID settings. To tackle this problem, recent works attemp to improve the model update schemes or model structures, e.g., modifying the SGD momentum term (Lin et al., 2021), replacing batch normalization with layer normalization (Hsieh et al., 2020), updating on clustered local models (Khawatmi et al., 2017), or modifying model update direction for personalized tasks (Esfandiari et al., 2021). Another line of works directly studies the effects of communication topology on consensus rates (Huang et al., 2022; Yuan et al., 2022; Song et al., 2022; Vogels et al., 2022). Comparing to DLPM, these methods still focus on achieving a global consensus model rather than optimizing personalized models for local tasks. In addition, comparing to the cooperation graph in SCooL, their mixing weights are usually pre-defined instead of automatically optimized for local tasks. SPDB (Lu et al., 2022) learns a shared backbone with personalized heads for local tasks. However, sharing the same backbone across all tasks might be sub-optimal and they do not optimize the mixing weights for peer-to-peer cooperation.

3. Probabilistic Cooperative Learning

3.1. Probabilistic Modeling with Cooperation Graph

We study a probabilistic model whose posterior probability of local models $\theta_{1:K}$ given local data $D_{1:K}$ is defined by

$$P(\theta_{1:K}|D_{1:K}) \propto P(\theta_{1:K}, D_{1:K}) = \int P(D_{1:K}|\theta_{1:K}, Y)P(\theta_{1:K}, Y)dY. \quad (1)$$

The cooperative learning of $\theta_{1:K}$ aims to maximize the posterior. Different from conventional decentralized learning methods like D-PSGD which fixes the cooperation graph or mixing weights, we explicitly optimize the cooperation graph Y for more effective cooperation among decentralized models maximizing $P(D_{1:K}|\theta_{1:K}, Y)$. The posterior in Eq. (1) is decomposed into two parts: the joint prior of $\theta_{1:K}$ and Y, and the joint likelihood of $D_{1:K}$ given $\theta_{1:K}$ and Y. By assuming different structures of the two parts (case 1.1-1.2 and case 2.1-2.3) and applying different priors $P(Y)$ for Y, we achieve a general framework from which we can derive a rich class of decentralized cooperative learning algorithms.

3.2. Configurations of Joint Likelihood and Prior

In the following, we will discuss several possible configurations of the general probabilistic model in Eq. (1).

Joint Likelihood $P(D_{1:K}|\theta_{1:K}, Y)$ Maximizing this joint likelihood optimizes both models θ and the cooperation graph Y to fit the datasets $D_{1:K}$. In a trivial case with Y fixed, it reduces to classical decentralized learning. In contrast, the joint likelihood allows us to choose a cooperation graph Y determining the data distributions of clients:

case 1.1 $P(D_{1:K}|\theta_{1:K})$: when Y is pre-defined without affecting the data distribution, the joint likelihood can be designed as a simple product of likelihoods over all clients.

$$P(D_{1:K}|\theta_{1:K}) = \prod_{i=1}^{K} P(D_i|\theta_i) \quad (2)$$

case 1.2 $P(D_{1:K}|\theta_{1:K}, Y)$ enables us to optimize the cooperation graph to coordinate the training of multiple local models. For example, the following joint likelihood model leads to a multi-task learning objective:

$$P(D_{1:K}|\theta_{1:K}, Y) = \prod_{i=1}^{K} P(D_{1:K}|\theta_i, Y) = \prod_{i=1}^{k} \left(P(D_i|\theta_i) \prod_{j \neq i, Y_{ij}=1} P(D_j|\theta_i) \right). \quad (3)$$

This objective leads to a personalized model θ_i learned from multiple sources of data $D_{1:K}$ and Y provides the mixing weights for different sources: $Y_{ij} = 1$ encourage a cooperation between client-i and client-j so the learning of θ_i can benefit from learning an additional task on D_j; while $Y_{ij} = 0$ indicates that learning task-j's data D_j hardly bring improvement to θ_i on task-i. **Joint Priors of Personalized Models and Cooperation Graph** $P(\theta_{1:K}, Y)$ can be parameterized in three different forms by presuming different dependencies of models and cooperation graphs:

case 2.1 : $P(\theta_{1:K}|Y)P(Y)$, $\theta_{1:K}$ is derived from Y.
case 2.2 : $P(Y|\theta_{1:K})P(\theta_{1:K})$, $\theta_{1:K}$ determines Y.
case 2.3 : $P(\theta_{1:K})P(Y)$, $\theta_{1:K}$ is independent to Y.

By choosing a joint prior from **case 2.1-2.3** and combining it with a joint likelihood chosen from **case 1.1-1.2**, we are able to create a rich family of probabilistic models that relate local models through their cooperation graph. In particular, the cooperation graph Y can guide the cross-client cooperation by relating either different clients' data (case 1.2) or their models (case 2.1-2.2). Practical designs of likelihood and prior need to consider the feasibility and efficiency of inference.

The generation of cooperation graph Y in the probabilistic model plays an important role in determining knowledge transfer across clients in cooperative learning. As shown in Section 4, if clients' tasks have a clustering structure and the clients belonging to the same cluster have a higher probability to cooperate, we can choose a stochastic block model (SBM) $P(Y)$ as the prior to generating Y; if we encourage the cooperation between clients with similar tasks or models, we can generate Y_{ij} via $P(Y_{ij}|\theta_i, \theta_j)$ according to the similarity between θ_i and θ_j, which can be captured by an "attention prior" that will be introduced later.

3.3. Variational Inference of Cooperation Graph & Cooperative Learning of Personalized Models

Maximizing the posterior in Eq. (1) requires an integral on latent variable Y, which may not have a closed form or is expensive to compute by sampling methods. Hence, we choose to use variational inference with mean field approximation (Jordan et al., 1999) to derive an EM algorithm that alternately updates the cooperation graph and local models using efficient closed-form updating rules. Despite possible differences in the concrete forms of likelihood and prior, the derived EM algorithm for different probabilistic models shares the same general form below.

For observations X (e.g., $X = D_{1:K}$), the set of all latent variables Z (e.g., $Z \supseteq Y$), and the set of all model parameters Φ (e.g., $\Phi \supseteq \theta_{1:K}$), the posterior is lower bounded by

$$\log p(X|\Phi) = \log \int p(X, Z|\Phi)dZ \geq \int q(Z) \log \frac{p(X, Z|\Phi)}{q(Z)} dZ := H(q, \Phi). \quad (4)$$

EM algorithm aims to maximize $L(q, \Phi)$ by iterating between the following E-step and M-step:

E-step finds distribution q to maximize the lower bound:

$$q \leftarrow \arg\max_{q} H(q, \Phi). \tag{5}$$

However, directly optimizing q is usually intractable. Hence, we resort to mean-field theory that approximates $q(Z)$ by a product distribution with variational parameter β_i for each latent variable Z_i, i.e.,

$$q_\beta(Z) = \prod_i q_i(Z_i|\beta_i). \tag{6}$$

Then the E-step reduces to:

$$\beta \leftarrow \arg\max_{\beta} H(q_\beta, \Phi). \tag{7}$$

For SCooL, its E-step has a closed-form solution $F(\cdot)$ to the variational parameters w of the cooperation graph Y so its E-step has the form of

$$w_{ij} \leftarrow F\Big(\log P(D_j|\theta_i), \beta, \Phi\Big)\ \forall i, j \in [K]. \tag{8}$$

Update other variational parameters in $\{\beta\}\backslash\{w_{ij}\}$.

M-step optimizes parameters Φ given the updated q, i.e.,

$$\Phi \leftarrow \arg\max_{\Phi} H(q, \Phi). \tag{9}$$

For SCooL, its M-step applies gradient descent to optimize the local models $\theta_{1:K}$.

$$\theta_i \leftarrow \theta_i - \eta_1\Big(\sum_{j\neq i} w_{ij}\nabla L(D_j;\theta_i) + \nabla L(D_i;\theta_i) + G(\beta,\Phi)\Big).$$

Update other observable variables $\{\Phi\}\backslash\{\theta_i\}$, (10)

where $L(D;\theta)$ is the loss of model θ computed on dataset D and $G(\beta, \Phi)$ is a term that only depends on β and Φ.

Remarks: In E-step shown in Eq. (8), SCooL updates w_{ij} based on the "cross-client loss" $\log P(D_j|\theta_i)$ that evaluates the personalized model θ_i of client-i on the dataset D_j of client-j. Intuitively, a higher log-likelihood $\log P(D_j|\theta_i)$ implies that the tasks on client-i and client-j are similar so θ_i can be improved by learning from θ_j via a larger w_{ij} (or Y_{ij}) in the cooperation graph. In M-Step shown in Eq. (10), SCooL trains each personalized model θ_i by not only using its own gradient $\nabla L(D_i;\theta_i)$ but also aggregating the gradients $\nabla L(D_j;\theta_i)$ computed on other clients with **mixing weights** w_{ij} from the cooperation graph. This encourages cooperative learning among clients with similar tasks. In Appendix C.1, we discuss a practical approximation to $\nabla L(D_j;\theta_i)$ that avoids data sharing between clients and saves communication cost without degrading cooperative learning performance.

Therefore, by iterating between E-step and M-step for SCooL, we optimize the cooperation graph to be adaptive to the local training progress on clients and their latest personalized models, which are then updated via cooperative learning among relevant clients on the optimized graph.

4. Graphical Model Priors for Cooperation Graph & Three Instantiations of SCooL

Algorithm 1: **S**tructured **Coo**perative **L**earning

Input $\{D_i\}_{i=1}^K, S, T$
Output $\theta_{1:K}$
Initialize personalized models $\theta_{1:K}$, latent variational parameters β, observable variables Φ
for $t = 0 \rightarrow T$ **do**
  for *client* $i = 1 \rightarrow K$ ***in parallel*** **do**
    E-step:
    $w_{ij} \leftarrow F\Big(\log P(D_j|\theta_i), \beta, \Phi\Big)$.
    Update $\{\beta\}\backslash\{w_{ij}\}$.
    Examples:
      SCooL-SBM: Eq. (18)-(20)
      SCooL-attention: Eq. (29)
    M-step:
    for *local SGD step* $m = 0 : s$ **do**
      $\theta_i \leftarrow \theta_i - \eta_1\Big(\nabla L(D_i;\theta_i) + \sum_{j\neq i} w_{ij}\nabla L(D_j;\theta_i) + G(\beta,\Phi)\Big)$.
    end
    Update $\{\Phi\}\backslash\{\theta_i\}$.
    Examples:
      SCooL-SBM: Eq. (22)-(24)
      SCooL-attention: Eq. (30)-(31)
  end
end

In this section, we derive three instantiations of SCooL algorithms associated with three graphical model priors $P(Y)$ used to generate the cooperation graph Y in the probabilistic model. We first show that D-PSGD (Lian et al., 2017) can be derived as a special case of SCooL when applying a Dirac Delta prior to Y. We then derive SCooL-SBM and SCooL-attention that respectively use stochastic block model (SBM) and an "attention prior" to generate a structured Y. The probabilistic models for these three SCooL examples are shown in Fig.2.

4.1. Dirac Delta Prior Leads to D-PSGD

We choose prior $P(Y)$ as a simple Dirac Delta distribution and define $P(\theta|Y)$ as a manifold prior (Belkin et al., 2006) based on the pairwise distance between local models:

$$Y \sim \delta(w) \tag{11}$$

$$P(\theta_{1:K}|Y) \propto \exp(-\frac{\lambda}{2}\sum_{1\leq i,j\leq K} Y_{ij}||\theta_i - \theta_j||^2) \tag{12}$$

$$P(D_{1:K}|\theta_{1:K}) \propto \prod_{i=1}^{K} P(D_i|\theta_i). \tag{13}$$

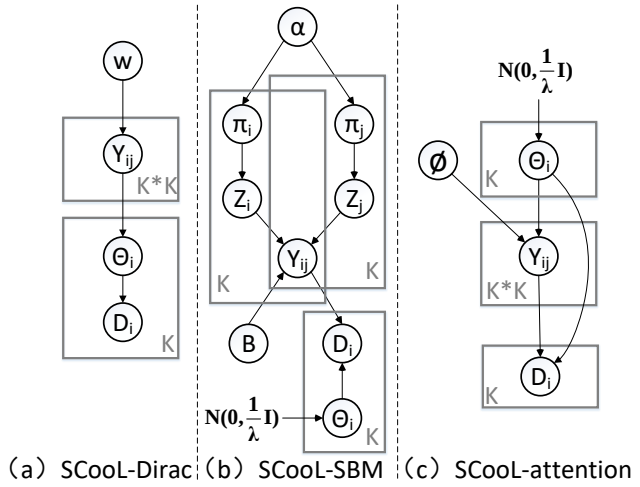


Figure 2. Probabilistic models used in SCooL examples.

We then choose **case 1.1** as the likelihood and **case 2.1** as the prior. Hence, maximizing posterior or MAP is:

$$\begin{aligned}&\arg\max_{\theta_{1:K}} \quad \log P(\theta_{1:K}|D_{1:K})\\ =&\arg\min_{\theta_{1:K}} \quad \sum_{i=1}^{K} L(D_i;\theta_i) + \frac{\lambda}{2}\sum_{i,j} w_{ij}||\theta_i-\theta_j||^2. \quad (14)\end{aligned}$$

The above MAP can be addressed by gradient descent:

$$\begin{aligned}\theta_i &\leftarrow \theta_i - \alpha\left(\nabla_\theta L(D_i;\theta_i) + \frac{\lambda}{2}\sum_{j=1}^{K}(w_{ij}+w_{ji})(\theta_i-\theta_j)\right)\\ &\overset{①}{=} \theta_i - \alpha\left(\nabla_\theta L(D_i;\theta_i) + \lambda\sum_{j=1}^{K} w_{ij}(\theta_i-\theta_j)\right)\\ &\overset{②}{=} \theta_i - \alpha\nabla_\theta L(D_i;\theta) - \alpha\lambda\left(\theta_i - \sum_{j=1}^{K} w_{ij}\theta_j\right), \quad (15)\end{aligned}$$

where ① holds because we enforce $w_{ij} = w_{ji}$, and ② holds due to the constraint $\sum_{j=1}^{K} w_{ij} = 1$.

Taking $\lambda = \frac{1}{\alpha}$, we finally obtain the update rule as:

$$\theta_i \leftarrow \sum_{j=1}^{K} w_{ij}\theta_j - \alpha\nabla_\theta L(D_i;\theta_i) \quad (16)$$

Hence, it exactly reconstruct the D-PSGD (Lian et al., 2017) algorithm (See Appendix B for details of D-PSGD).

4.2. SCooL-SBM with Stochastic Block Model Prior

In cooperative learning, we can assume a clustering structure of clients such that the clients belonging to the same community benefit more from their cooperation. This structure can be captured by stochastic block model (SBM) (Holland et al., 1983), which is used as the prior generating the cooperation graph in SCooL. In particular,

- For each client $i \in [K]$:
  - Draw an M-dimensional membership probability distribution as a vector $\vec{\pi}_i \sim \text{Dirichlet}(\vec{\alpha})$.
  - Draw membership label $\vec{z}_i \sim \text{Multinomial}(\vec{\pi}_i)$.
- For each pair of clients $(i,j) \in [K]\times[K]$:
  - Sample $Y_{ij} \sim \text{Bernoulli}(\vec{z}_i^T B\ \vec{z}_j)$ that determines the cooperation between client pair (i,j).

Hence, the marginal distribution of Y under SBM is:

$$P(Y|\vec{\alpha},B) = \int P(Y,\vec{\pi}_{1:K},\vec{Z}_{1:K}|\vec{\alpha},B)d(\vec{\pi}_{1:K},\vec{Z}_{1:K}).$$

We assume Gaussian priors for personalized models:

$$P(\theta_{1:K}) \propto \exp(-\frac{\lambda}{2}\sum_i ||\theta_i||^2). \quad (17)$$

SCooL-SBM: Since the generation of Y does not depend on θ, we can consider the joint prior in **case 2.3**. We further choose **case 1.2** as the likelihood. The EM algorithm for SCooL-SBM can be derived as the following (details are given in Appendix D).

- **E-step** updates w, γ, and Ω, which are the variaiontal parameters of latent variables Y, π, and z, respectively.

$$\begin{aligned}w_{ij} \leftarrow \text{Sigmoid}\Big(&\log P(D_j|\theta_i) + \sum_{g,h}\vec{\Omega}_{ig}\vec{\Omega}_{jh}\log\\ &B(g,h) - \sum_{g,h}\vec{\Omega}_{ig}\vec{\Omega}_{jh}\log(1-B(g,h))\Big) \quad (18)\end{aligned}$$

$$\gamma_{ig} \leftarrow \vec{\Omega}_{ig} + \vec{\alpha}_g \quad (19)$$

$$\begin{aligned}\vec{\Omega}_{i\cdot} \leftarrow \text{Softmax}\Big(&\sum_j w_{ij}\sum_h \vec{\Omega}_{jh}\log B(\cdot,h) +\\ &\sum_j w_{ji}\sum_h \vec{\Omega}_{jh}\log B(h,\cdot) \quad (20)\\ &+\psi(\gamma_{i\cdot}) - \psi\left(\sum_g \gamma_{ig}\right) +\\ &\sum_j(1-w_{ij})\sum_h \vec{\Omega}_{jh}\log(1-B(\cdot,h)) +\\ &\sum_j(1-w_{ji})\sum_h \vec{\Omega}_{jh}\log(1-B(h,\cdot))\Big) \quad (21)\end{aligned}$$

- **M-step** updates the model parameters θ_i, $\vec{\alpha_i}$, and B.

$$\theta_i \leftarrow \theta_i - \eta_1 \Bigg(\nabla_{\theta_i} L(D_i; \theta_i) + \sum_{j \neq i} w_{ij} \nabla_{\theta_i} L(D_j; \theta_i) + \lambda \theta_i \Bigg) \tag{22}$$

$$\vec{\alpha_i} \leftarrow \vec{\alpha_i} + \eta_2 \Bigg(\sum_g \Bigg(\psi(\gamma_{gi}) - \psi(\sum_k \gamma_{gk}) \Bigg) - K \sum_g \psi(\vec{\alpha}_g) + K \psi \Bigg(\sum_k \vec{\alpha}_k \Bigg) \Bigg) \tag{23}$$

$$B(i,j) \leftarrow \frac{\sum_{g,h} w_{gh} \vec{\Omega}_{gi} \vec{\Omega}_{hj}}{\sum_{g,h} \vec{\Omega}_{gi} \vec{\Omega}_{hj}} \tag{24}$$

4.3. SCooL-attention with Attention Prior

Instead of assuming a clustering structure, we can train an attention mechanism to determine whether every two clients can benefit from their cooperation. In the probabilistic model of SCooL, we can develop an attention prior to generate the cooperation graph. Specifically, we compute the attention between between θ_i and θ_j using a learnable metric $f(\cdot, \cdot)$, i.e.,

$$p_{ij} = \frac{\exp(f(\theta_i, \theta_j))}{\sum_l \exp(f(\theta_i, \theta_\ell))}, \tag{25}$$

In dot-product attention, $f(\theta_i, \theta_j)$ is the inner product between the representations of θ_i and θ_j produced by a learnable encoder $E(\cdot; \phi)$, e.g., the representation of θ_i is computed as $E(\theta_i^{t+\frac{1}{2}} - \theta_i^0; \phi)$. We compute the difference $\theta_i^{t+\frac{1}{2}} - \theta_i^0$ in order to weaken the impact of initialization θ_i^0 and focus on the model update produced by gradient descent in the past training rounds. Hence,

$$f(\theta_i, \theta_j) = \langle E(\theta_i^{t+\frac{1}{2}} - \theta_i^0; \phi), E(\theta_j^{t+\frac{1}{2}} - \theta_j^0; \phi) \rangle. \tag{26}$$

Each row of Y is a one-hot vector drawn from a categorical distribution defined by the attention scores between personalized models, i.e.,

$$\vec{Y_i} \sim \text{Categorical}(p_{i1}, p_{i2}, ..., p_{iK}), \ \forall \ i \in [K] \tag{27}$$

In SCooL-attention's probabilistic model, we also use Gaussian as the prior for personalized models:

$$P(\theta_{1:K}) \propto \exp(-\frac{\lambda}{2} \sum_i ||\theta_i||^2), \tag{28}$$

SCooL-attention Hence, the above defines a joint prior in the form of **case 2.2**. We further adopt the likelihood in **case 1.2**. The EM algorithm for SCooL-attention can then be derived as the following (details given in Appendix E).

- **E-step** updates the variational parameters w of cooperation graph Y, i.e.,

$$w_{i\cdot} \leftarrow \text{Softmax} \Big(\log P(D_\cdot | \theta_i) + \log p_{i\cdot} \Big) \tag{29}$$

- **M-step** upates the model parameters θ_i and ϕ, i.e.,

$$\theta_i \leftarrow \theta_i - \eta_1 \Bigg(\nabla_{\theta_i} L(D_i; \theta_i) + \sum_{j \neq i} w_{ij} \nabla_{\theta_i} L(D_j; \theta_i) + \lambda \theta_i - \sum_{ij} w_{ij} \nabla_{\theta_i} \log p_{ij} \Bigg) \tag{30}$$

$$\phi \leftarrow \phi + \eta_2 \nabla_\phi \Bigg(\sum_{ij} w_{ij} \log p_{ij} \Bigg) \tag{31}$$

5. Experiments

Table 1. The parameters of both non-IID (McMahan et al., 2017) and non-IID SBM experimental setup.

Dataset	M	K	N	model
CIFAR-10	10	100	2	two-layer CNN
CIFAR-100	100	100	10	two-layer CNN
MiniImageNet	100	100	10	four-layer CNN

5.1. Experimental Setup

To test the personalization performances of SCooL models, we draw classification tasks from two non-IID settings:

- **Non-IID SBM**: a simpler non-IID setting. Given a dataset of M classes, the totally K clients can be divided into several groups. Clients within the same group are allocated with the same subset of N classes, while clients from different groups do not have any shared classes. This task assignment distribution among clients can be described by a SBM model with an uniform membership prior α.
- **Non-IID** (McMahan et al., 2017): A more challenging non-IID setting used in FedAvg (McMahan et al., 2017), where we randomly assign N classes of data to every client from totally M classes. Different from non-IID SBM setting, no pair of clients is ensured to share the same classes and data distribution.

We evaluate SCooL-SBM and SCooL-attention and compare them with several FL/DL baselines on three datasets: **CIFAR-10** (Krizhevsky et al., 2009), **CIFAR-100**, and **MiniImageNet** (Ravi & Larochelle, 2017), each having 50,000 training images and 10,000 test images. In Table 3,

we list the parameters for the two non-IID settings. Following the evaluation setting for personalized models in previous non-IID DL works (Liang et al., 2020; Zhang et al., 2021), we evaluate each local model on all available test samples belonging to the classes in its local task. We run every experiment using five different random seeds and report their average test accuracy. We choose local models to be the two-layer CNN adopted in FedAvg (McMahan et al., 2017) for CIFAR-10/100 and the four-layer CNN adopted in MAML (Finn et al., 2017) for MiniImageNet. Since BatchNorm (BN) may have a detrimental effect on DL (Hsieh et al., 2020), we replace all the BN layers (Ioffe & Szegedy, 2015) with group-norm layers (Wu & He, 2018). The implementation details of SCooL-SBM and SCooL-attention are given in Appendix C.1.

Baselines We compare our methods with a diverse set of baselines from federated learning (FL) and decentralized learning (DL) literature, as well as a **local SGD only baseline** without any model aggregation across clients. FL baselines include **FedAvg** (McMahan et al., 2017) (the most widely studied FL method), **Ditto** (Li et al., 2021a) achieving fairness and robustness via a trade-off between the global model and local objectives, and **FOMO** (Zhang et al., 2021) applying adaptive mixing weights to combine neighbors' models for updating personalized models. DL baselines include **D-PSGD** (Lian et al., 2017) with fixed mixing weights and topology, **CGA** (Esfandiari et al., 2021) with a fixed topology but adaptive mixing weights for removing the conflict of cross-client gradients, **SPDB** (Lu et al., 2022) with a shared backbone network but personalized heads for clients' local tasks, **meta-L2C** (Li et al., 2022) learning mixing weights to aggregate clients' gradients, and **Dada** (Zantedeschi et al., 2020) training local models with weighted regularization to the pairwise distance between local models.

We run each baseline for 100 (communication) rounds or equally 500 local epochs if > 100 rounds are needed, the same as our methods, except FedAvg which needs more (i.e., > 1000) epochs to converge. For fair comparisons, we keep their communication cost per client and local epochs in each round to be no smaller than that of our methods. For FL baselines, the communication happens between the global server and clients, so we randomly select 10% clients for aggregation and apply 5 local epochs per client in each round. For DL baselines, we let every client communicate with $\sim$10% clients in every round. we evaluate them in two settings, i.e., one local SGD step per round and 5 local epochs per round. We evaluate each baseline DL method on multiple types of communication topology and report the best performance. More details are provided in Appendix C.2.

Training hyperparameters In all methods' local model training, we use SGD with learning rate of 0.01, weight decay of 5×10^{-4}, and batch size of 10. We follow the hyperparameter values proposed in the baselines' papers except the learning rate, which is a constant tuned/selected from $[0.01, 0.05, 0.1]$ for the best validation accuracy.

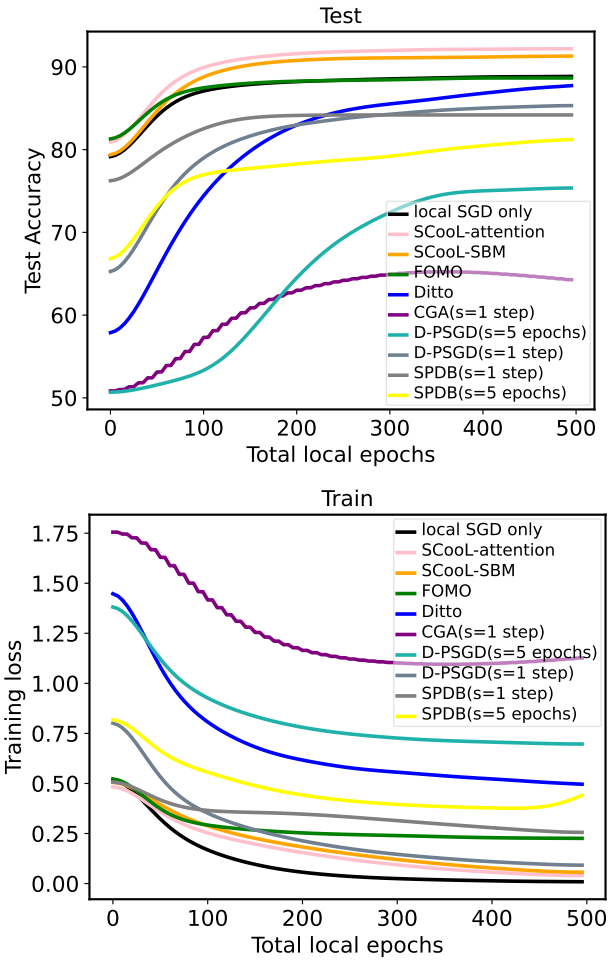


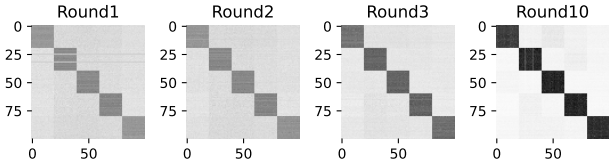
Figure 3. **Test accuracy and training loss vs. total local epochs** on CIFAR-10. SCooL-SBM and SCooL-attention converge faster to better test/training performance than FL/DL baselines. FedAvg requires > 1000 epochs to converge and is not included.

5.2. Experimental Results

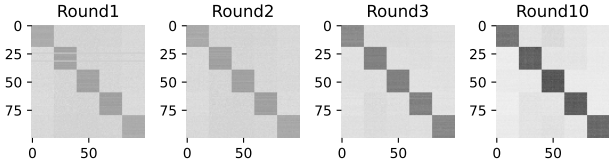
Test accuracy and convergence Table 2 reports the test accuracy of all the 100 clients' models on their assigned non-IID tasks (mean$\pm$std over all clients). SCooL-SBM and SCooL-attention outperform FL/DL baselines by a large margin on all the three datasets. Moreover, SCooL-attention's prior can capture the pairwise similarity between clients in the non-IID setting so it outperforms SCooL-SBM. On the other hand, in the non-IID SBM setting, SCooL-SBM outperforms SCooL-attention since SCooL-SBM's prior is a better model of the SBM generated cooperation graph. In Fig. 3, we compare the convergence of test accuracy and training loss for all methods in the non-IID setting, where SCooL-SBM and SCooL-attention converge faster than others.

Table 2. Test accuracy (mean±std) of 100 local models for non-IID tasks. SCooL-SBM and SCooL-attention outperform FL/DL baselines.

Methodology	Algorithm	non-IID (McMahan et al., 2017)			non-IID SBM		
		CIFAR-10	CIFAR-100	MiniImageNet	CIFAR-10	CIFAR-100	MiniImageNet
Local only	Local SGD only	87.5±7.02	55.47±5.20	41.59±7.71	87.41±4.21	55.37±3.48	38.54±7.94
Federated	FedAvg	70.65±10.64	40.15±7.25	34.26±6.01	71.59±12.85	39.89±11.42	38.87±9.72
	FOMO	88.72±5.41	52.44±5.09	44.56±4.31	90.30±2.67	67.31±4.81	42.72±2.23
	Ditto	87.32±6.42	54.28±5.31	42.73±5.19	88.13±7.43	54.34±5.42	42.16±5.46
Decentralized	D-PSGD(s=1 step)	83.01±7.34	40.56±6.94	30.26±5.75	85.20±4.05	48.15±4.77	37.43±3.59
	D-PSGD(s=5 epochs)	75.89±6.65	35.03±4.83	28.41±5.18	77.33±5.79	32.17±5.07	37.69±3.02
	CGA(s=1 step)	65.65±12.66	30.81±10.79	27.65±11.78	69.93±5.34	36.91±7.58	25.54±1.95
	CGA(s=5 epochs)	diverge	diverge	diverge	diverge	diverge	diverge
	SPDB(s=1 step)	82.36±7.14	54.29±6.15	39.17±3.93	81.75±7.07	55.71±6.02	38.49±5.12
	SPDB(s=5 epochs)	81.15±7.06	53.23±7.48	35.93±5.05	81.25±6.07	53.08±4.01	35.86±4.03
	Dada	85.65±6.36	57.61±5.45	37.81±7.15	88.89±3.47	64.62±4.77	41.68±3.91
	meta-L2C	92.10±4.71	58.28±3.09	48.80±4.17	91.84±2.40	71.64±2.89	49.95±1.97
SCooL(Ours)	SCooL-SBM	91.37±5.03	58.76±4.30	48.69±5.21	**94.14±2.28**	**72.27±2.59**	**51.86±1.64**
	SCooL-attention	**92.21±5.15**	**59.47±4.95**	**49.53±3.29**	93.98±3.85	72.03±2.71	51.69±2.80

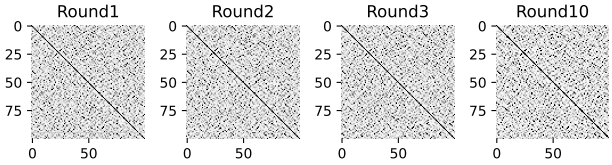


(a) SCooL-SBM on non-IID SBM setting.

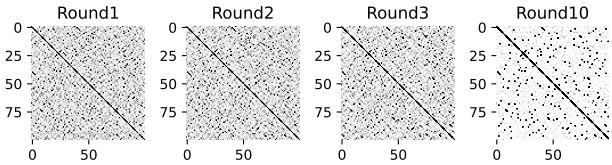


(b) SCooL-attention on non-IID SBM setting.

Figure 4. Cooperation graph weights w by SCooL-SBM and SCooL-attention applied to the **non-IID SBM** setting. Both SCooL-SBM and SCooL-attention capture the ground-truth task similarity even in earlier training states, while the cooperation graphs of SCooL-SBM converge faster in non-IID SBM setting.



(a) SCooL-SBM on non-IID (McMahan et al., 2017) setting.



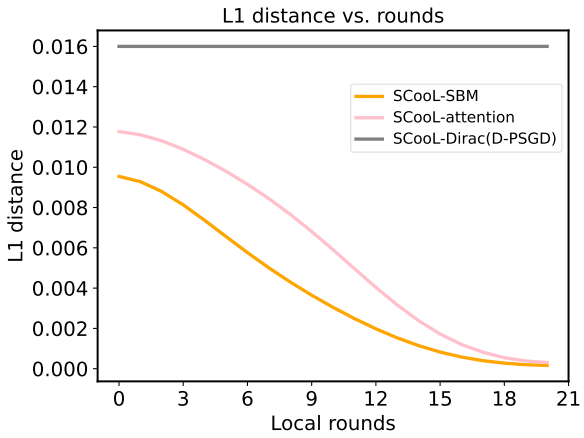
(b) SCooL-attention on non-IID (McMahan et al., 2017) setting.

Figure 5. Cooperation graph weights w for 100 clients by SCooL-SBM and SCooL-attention applied to **non-IID** setting (McMahan et al., 2017). Both SCooL-SBM and SCooL-attention generate sparse cooperation graphs after a few rounds. SCooL-attention's cooperation graph converge faster than SCooL-SBM.

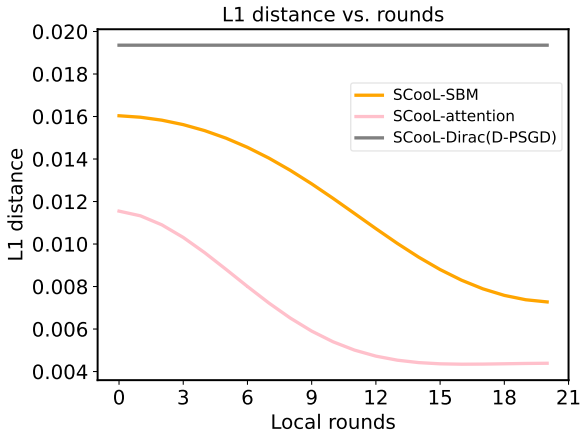
Learned cooperation graphs In Fig. 4a-4b, we report how the cooperation graphs produced by SCooL-SBM and SCooL-attention over communication rounds for non-IID SBM setting. Both methods capture the true task relationships after only a few training rounds. The faster convergence of SCooL-SBM indicates that SCooL-SBM's prior is a better model capturing the SBM cooperation graph structure. In Fig. 5a-5b, we report the learned mixing weights in the non-IID setting. Both algorithms can quickly learn a sparse cooperation graph very early, which significantly reduces the communication cost for later-stage training. In this setting, SCooL-attention is better and faster than SCooL-SBM on capturing the peer-to-peer correlations.

In addition, we conduct a quantitative evaluation to the learned cooperation graphs by comparing the mixing weights w with the ground truth w^* used to draw the non-IID tasks. In the ground truth w^*, $w^*_{ij} = 1$ for two clients i and j sharing the same data distribution and $w^*_{ij} = 0$ otherwise. We normalize each row in w^* so entries in each row sum up to one. Fig. 6a shows that the mixing weights in SCooL-SBM converge faster to the ground-truth than SCooL-attention and achieve a similar L1 error because SCooL with SBM prior can better capture SBM-generated graph structures in the non-IID SBM setting. In Fig. 6b, SCooL-attention is faster on finding a more accurate cooperation graph due to its attention prior modeling peer-to-peer similarity in the non-IID setting.

Communication cost/budget The communication cost is often a bottleneck of DL so a sparse topology is usually preferred in practice. In Fig. 7, we evaluate the personalization performance of SCooL-SBM, SCooL-attention, and D-PSGD under different communication budgets. Both of our methods achieve almost the same accuracy under different budgets while D-PSGD's performs much poorer and its accuracy highly depends on an increased budget. In contrast, SCooL only requires communication to 2% neighbours on average to achieve a much higher test accuracy.



(a) Non-IID SBM setting.



(b) Non-IID setting.

Figure 6. Mixing weights L1 distance to ground-truth Y in the first 20 training rounds out of totally 100 rounds, (a) for non-IID SBM setting, (b) for non-IID (McMahan et al., 2017) setting. Both SCooL-SBM and SCooL-attention finally converge to generate more precise mixing weights. SCooL-SBM adapts mixing weights faster on non-IID SBM setting, while SCooL-attention adapts faster and converges to lower L1 error in non-IID setting.

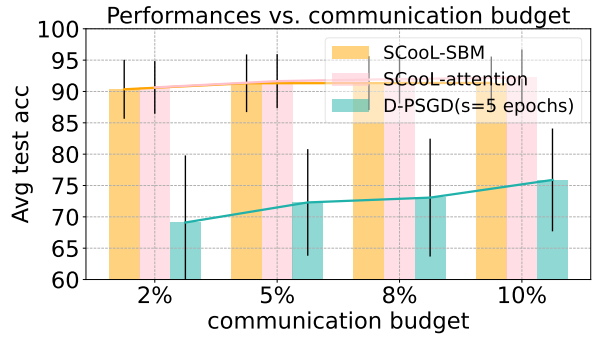


Figure 7. SCooL-SBM and SCooL-attention are robust to communication cost/budget changes. We evaluate each method for every client communicating with 2%, 5%, 8%, or 10% of other clients.

6. Conclusion

We propose a probabilistic modeling scheme"Structured Cooperative Learning (SCooL)" for decentralized learning of personalized models. SCooL improves the cooperative learning of personalized models across clients by alternately optimizing a cooperation graph Y and the personalized models. We introduce three instantiations of SCooL that adopt different graphical model priors to generate Y. They leverage the structural prior among clients to capture an accurate cooperation graph that improves each local model by its neighbors' models. We empirically demonstrate the advantages of SCooL over SOTA Federated/Decentralized Learning methods on personalization performance and communication efficiency in different non-IID settings. SCooL is a general framework for efficient knowledge sharing between decentralized agents on a network. It combines the strengths of both the neural networks on local data fitting and graphical models on capturing the cooperation graph structure, leading to interpretable and efficient decentralized learning algorithms with learnable cooperation. In the future work, we are going to study SCooL for partial-model personalization and other non-classification tasks.

Acknowledgement

Shuangtong and Prof. Xinmei Tian are partially supported by NSFC No. 62222117 and the Fundamental Research Funds for the Central Universities under contract WK3490000005. Prof. Dacheng Tao is partially supported by Australian Research Council Project FL-170100117.

References

Acar, D. A. E., Zhao, Y., Matas, R., Mattina, M., Whatmough, P., and Saligrama, V. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=B7v4QMR6Z9w.

Afonin, A. and Karimireddy, S. P. Towards model agnostic federated learning using knowledge distillation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=lQI_mZjvBxj.

Balakrishnan, R., Li, T., Zhou, T., Himayat, N., Smith, V., and Bilmes, J. Diverse client selection for federated learning via submodular maximization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nwKXyFvaUm.

Belkin, M., Niyogi, P., and Sindhwani, V. Manifold regularization: A geometric framework for learning from labeled

and unlabeled examples. *Journal of machine learning research*, 7(11), 2006.

Blot, M., Picard, D., Cord, M., and Thome, N. Gossip training for deep learning. *arXiv preprint arXiv:1611.09726*, 2016.

Chen, F., Long, G., Wu, Z., Zhou, T., and Jiang, J. Personalized federated learning with structural information. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2022.

Chen, H.-Y. and Chao, W.-L. Fed{be}: Making bayesian model ensemble applicable to federated learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=dgtpE6gKjHn.

Collins, L., Hassani, H., Mokhtari, A., and Shakkottai, S. Exploiting shared representations for personalized federated learning. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2089–2099. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/collins21a.html.

Dai, R., Shen, L., He, F., Tian, X., and Tao, D. Dispfl: Towards communication-efficient personalized federated learning via decentralized sparse training. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 4587–4604. PMLR, 2022. URL https://proceedings.mlr.press/v162/dai22b.html.

Esfandiari, Y., Tan, S. Y., Jiang, Z., Balu, A., Herron, E., Hegde, C., and Sarkar, S. Cross-gradient aggregation for decentralized learning from non-iid data. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3036–3046. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/esfandiari21a.html.

Fallah, A., Mokhtari, A., and Ozdaglar, A. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33, 2020.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135. PMLR, 2017.

Fraboni, Y., Vidal, R., Kameni, L., and Lorenzi, M. Clustered sampling: Low-variance and improved representativity for clients selection in federated learning. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3407–3416. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/fraboni21a.html.

Ghosh, A., Chung, J., Yin, D., and Ramchandran, K. An efficient framework for clustered federated learning. *arXiv preprint arXiv:2006.04088*, 2020.

Holland, P. W., Laskey, K. B., and Leinhardt, S. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

Hsieh, K., Phanishayee, A., Mutlu, O., and Gibbons, P. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pp. 4387–4398. PMLR, 2020.

Huang, Y., Sun, Y., Zhu, Z., Yan, C., and Xu, J. Tackling data heterogeneity: A new unified framework for decentralized SGD with sample-induced topology. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9310–9345. PMLR, 2022. URL https://proceedings.mlr.press/v162/huang22i.html.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/ioffe15.html.

Jiang, Z., Balu, A., Hegde, C., and Sarkar, S. Collaborative deep learning in fixed topology networks. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/a74c3bae3e13616104c1b25f9da1f11f-Paper.pdf.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Mach. Learn.*, 37(2):183–233, 1999. doi: 10.

1023/A:1007665907178. URL https://doi.org/10.1023/A:1007665907178.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Khawatmi, S., Sayed, A. H., and Zoubir, A. M. Decentralized clustering and linking by networked agents. *IEEE Transactions on Signal Processing*, 65(13):3526–3537, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Li, S., Zhou, T., Tian, X., and Tao, D. Learning to collaborate in decentralized learning of personalized models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

Li, T., Hu, S., Beirami, A., and Smith, V. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pp. 6357–6368. PMLR, 2021a.

Li, X., JIANG, M., Zhang, X., Kamp, M., and Dou, Q. FedBN: Federated learning on non-IID features via local batch normalization. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=6YEQUn0QICG.

Lian, X., Zhang, C., Zhang, H., Hsieh, C.-J., Zhang, W., and Liu, J. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/f75526659f31040afeb61cb7133e4e6d-Paper.pdf.

Liang, P. P., Liu, T., Ziyin, L., Allen, N. B., Auerbach, R. P., Brent, D., Salakhutdinov, R., and Morency, L.-P. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

Lin, T., Kong, L., Stich, S. U., and Jaggi, M. Ensemble distillation for robust model fusion in federated learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 2351–2363. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/18df51b97ccd68128e994804f3eccc87-Paper.pdf.

Lin, T., Karimireddy, S. P., Stich, S., and Jaggi, M. Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6654–6665. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/lin21c.html.

Long, G., Xie, M., Shen, T., Zhou, T., Wang, X., and Jiang, J. Multi-center federated learning: clients clustering for better personalization. *World Wide Web Journal (Springer)*, 2022.

Lu, S., Cui, X., Squillante, M. S., Kingsbury, B., and Horesh, L. Decentralized bilevel optimization for personalized client learning. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5543–5547. IEEE, 2022.

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

Oh, J., Kim, S., and Yun, S.-Y. FedBABU: Toward enhanced representation for federated image classification. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=HuaYQfggn5u.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017.

Sattler, F., Müller, K.-R., and Samek, W. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 2020.

Shamsian, A., Navon, A., Fetaya, E., and Chechik, G. Personalized federated learning using hypernetworks. In Meila, M. and Zhang, T. (eds.), *Pro-*

ceedings of the 38th International Conference on Machine Learning, volume 139 of *Proceedings of Machine Learning Research*, pp. 9489–9502. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/shamsian21a.html.

Song, Z., Li, W., Jin, K., Shi, L., Yan, M., Yin, W., and Yuan, K. Communication-efficient topologies for decentralized learning with $o(1)$ consensus rate. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=AyiiHcRzTd.

T. Dinh, C., Tran, N., and Nguyen, J. Personalized federated learning with moreau envelopes. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21394–21405. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/f4f1f13c8289ac1b1ee0ff176b56fc60-Paper.pdf.

Tan, Y., Long, G., Liu, L., Zhou, T., Lu, Q., Jiang, J., and Zhang, C. Fedproto: Federated prototype learning across heterogeneous clients. In *AAAI Conference on Artificial Intelligence*, volume 1, 2022a.

Tan, Y., Long, G., Ma, J., Liu, L., Zhou, T., and Jiang, J. Federated learning from pre-trained models: A contrastive learning approach. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022b. URL https://openreview.net/forum?id=mhQLcMjWw75.

Vogels, T., Hendrikx, H., and Jaggi, M. Beyond spectral gap: the role of the topology in decentralized learning. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=AQgmyyEWg8.

Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., and Khazaeni, Y. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BkluqlSFDS.

Wu, Y. and He, K. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Xie, M., Long, G., Shen, T., Zhou, T., Wang, X., Jiang, J., and Zhang, C. Multi-center federated learning. *arXiv preprint arXiv:2108.08647*, 2021.

Xu, C., Hong, Z., Huang, M., and Jiang, T. Acceleration of federated learning with alleviated forgetting in local training. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=541PxiEKN3F.

Yuan, K., Huang, X., Chen, Y., Zhang, X., Zhang, Y., and Pan, P. Revisiting optimal convergence rate for smooth and non-convex stochastic decentralized optimization. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=eHePKMLuNmy.

Zantedeschi, V., Bellet, A., and Tommasi, M. Fully decentralized joint learning of personalized models and collaboration graphs. In *International Conference on Artificial Intelligence and Statistics*, pp. 864–874. PMLR, 2020.

Zhang, C., Long, G., Zhou, T., Yan, P., Zhang, Z., Zhang, C., and Yang, B. Dual personalization on federated recommendation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.

Zhang, M., Sapra, K., Fidler, S., Yeung, S., and Alvarez, J. M. Personalized federated learning with first order model optimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=ehJqJQk9cw.

Zhu, Z., Hong, J., and Zhou, J. Data-free knowledge distillation for heterogeneous federated learning. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12878–12889. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/zhu21b.html.

A. Notations

Table 3. Notations used in this paper.

Notation	Description
θ_i	Personalized model on the i'th client
D_i	Dataset for the i'th client
Y	cooperation graph
Φ	Observable variables set
Z	Latent variables set
β	Variational parameters of latent variables Z
w	Variational parameter of cooperation graph Y
π_i	Prior membership distribution of client-i in SBM model
α_i	Parameter for Dirichlet distribution of π_i
z_i	Membership indicator for client-i in SBM model
B	Pairwise correlations of memberships in SBM model
γ	Variational parameter of π
Ω	Variational parameter of z
ϕ	Learnable neural network parameters for attention prior

B. D-PSGD algorithm

Algorithm 1 D-PSGD

Require: initial point $x_{0,i} = x_0$, step length γ, weight matrix w, and number of iterations T
for $t = 0$ **to** $T - 1$ **do**
  Randomly sample $\zeta_{t,i}$ from local data of the i-th client
  Compute a local stochastic gradient based on $\zeta_{t,i}$ and current optimization variable $x_{t,i}$: $\nabla F_i(x_{t,i}; \zeta_{t,i})$
  Compute the neighborhood weighted average by fetching optimization variables from neighbors: $x_{t+\frac{1}{2},i} = \sum_{j=1}^{K} w_{ij} x_{t,j}$
  Update the local optimization variable $x_{t+1,i} \leftarrow x_{t+\frac{1}{2},i} - \gamma \nabla F_i(x_{t,i}; \zeta_{t,i})$
end for
Output: $\frac{1}{K} \sum_{i=1}^{K} x_{T,i}$

C. Experimental Details

C.1. Implementation Details of SCooL-SBM and SCooL-attention

We apply a lightweight fully connected network of two layers with output dimensions $(10, 5)$ as the encoder network ϕ in SCooL-attention. We use Adam (Kingma & Ba, 2014) with learning rate of 0.1 and weight decay of 0.01 to train both SCooL-SBM and SCooL-attention. We apply Algorithm 1 and train all the local models for T=100 rounds with 5 epochs of local SGD per round. To achieve a sparse topology, we sort the learned w_{i*} for each client, and remove 90% neighbors with the smallest w_{ij} for each client after 10 rounds.

The update rule of (22) and (30) requires local model θ_i to calculate gradients on other clients' dataset θ_j. When local data is not allowed to share across clients, we can follow a "cross-gradient" fashion: sending model θ_i to client-j, who then computes gradient on its own data and then sends $\nabla L(D_j; \theta_i)$ back to client-i. However, this method requires twice the communication cost of classical decentralized learning algorithms such as D-PSGD. To avoid such cross gradient terms, we can approximate $\nabla L(D_j; \theta_i)$ using $\nabla L(D_j; \theta_j)$ according to first-order Taylor expansion:

$$L(\theta_i; D_j) = L(\theta_j; D_j) + \nabla L(\theta_j; D_j)(\theta_i - \theta_j) + O(||\theta_i - \theta_j||^2) \tag{32}$$

When θ_i and θ_j are close to each other, i.e., $O(||\theta_i - \theta_j||^2)$ is small, we can ignore the second-order term and get an

approximation with small approximation error, i.e.,

$$\nabla_{\theta_i} L(\theta_i; D_j) \approx \nabla_{\theta_j} L(\theta_j; D_j). \tag{33}$$

In practice, if we initialize all personalized models from the same point, clients with similar or the same tasks tend to have similar optimization paths and thus their models' distance can be upper bounded. For clients with distinct tasks, the learned w_{ij} tends to be close to zero so the approximation error does not result in a notable difference in the model updates in M-step. In experiments, we find that the mixing weights converge quickly and precisely capture task relationships among clients. As shown in Table 2, this approximation already achieves promising personalization performances.

In practice, the training loss of $\log P(D_j|\theta_i)$ in equation (18)(29) can vary in magnitude in different training stages, causing the learned w "over-smooth" or "over-sharp" in certain epochs due to the nature of Sigmoid/ Sofrmax. To tackle this issue, we use Sigmoid/ Softmax with temperature factor. The temperature factor is kept fixed during the whole training phase and selected as a hyperparameter.

C.2. Details of Decentralized Learning Baselines

We train our SCooL models with a communication period of 5 epochs. Since the DL baselines, i.e. D-PSGD, CGA, SPDB, are originally proposed to only run one local SGD step per round on a single mini-batch, we evaluate them with two settings, i.e., one local step per round and 5 local epochs per round, and we apply more rounds for the former to match the total local epochs (i.e., 500 epochs) of other methods. To match the communication cost of our methods, we extend the ring and bipartite topology used in previous DL works (Esfandiari et al., 2021) to increase the number of neighbors for each client. Specifically, we study (1) a "group-ring" topology that connects two clients i and j if $|i-j| \leq \frac{(K-K_0)}{2}$ or $K-|i-j| \leq \frac{(K-K_0)}{2}$; and (2) a generalized bipartite topology that randomly partitions all clients into two groups and then connect each client in a group to 10 clients randomly drawn from the other group. In our experiments, they both outperform their original versions with fewer neighbors and communications. Hence, in the following, we always report the best result among all the four types of topology for each DL baseline.

D. SCooL-SBM derivation

$$\begin{aligned} P(\theta_{1:K}|D_{1:K}) &\propto P(\theta_{1:K}, D_{1:K}) \\ &= \int P(D_{1:K}|\theta_{1:K}, Y) P(\theta_{1:K}|Y) P(Y) dY \end{aligned} \tag{34}$$

with

$$\begin{aligned} &P(\theta_{1:K}, Y) = P(\theta_{1:K}) P(Y) \\ &P(Y) = \int P(Y, \vec{\pi}_{1:K}, \vec{Z}_{1:K}|\vec{\alpha}, B) d(\vec{\pi}_{1:K}, \vec{Z}_{1:K}) \\ &P(\theta_{1:K}) \propto \exp(-\frac{\lambda}{2} \sum_i ||\theta_i||^2) \\ &P(D_{1:K}|\theta_{1:K}, Y) = \prod_{i=1}^{K} P(D_{1:K}|\theta_i, Y) = \prod_{i=1}^{k} \left(P(D_i|\theta_i) \prod_{j \neq i, Y_{ij}=1} P(D_j|\theta_i) \right) \end{aligned} \tag{35}$$

D.0.1. OBJECTIVE

Modelling $P(Y)$ as SBM:

- For each client $i \in [K]$:
  - Draw a M dimensional membership vector $\vec{\pi}_i \sim \text{Dirichlet}(\vec{\alpha})$.
  - Draw membership indicator $\vec{z}_i \sim \text{Multinomial}(\vec{\pi}_i)$.
- For each pair of clients $(i, j) \in K \times K$:
  - Sample the value of their interaction, $Y_{ij} \sim \text{Bernoulli}(\vec{z}_i^T B \, \vec{z}_j)$.

Under the SBM, the marginal distribution of Y is:

$$\begin{aligned} P(Y|\vec{\alpha}, B) &= \int P(Y, \vec{\pi}_{1:K}, \vec{Z}_{1:K}|\vec{\alpha}, B) d(\vec{\pi}_{1:K}, \vec{Z}_{1:K}) \\ &= \int \Big(\prod_{ij} P(Y_{ij}|\vec{z}_i, \vec{z}_j, B) \prod_i P(\vec{\pi}_i|\vec{\alpha}_i) P(\vec{z}_i|\vec{\pi}_i) \Big) d(\vec{\pi}_{1:K}, \vec{Z}_{1:K}) \end{aligned} \tag{36}$$

Our final objective is:

$$\begin{aligned} & P(\theta_{1:K}|D_{1:K}) \propto P(\theta_{1:K}, D_{1:K}) \\ &= \int P(D_{1:K}|\theta_{1:K}, Y) P(\theta_{1:K}) P(Y) dY \\ &= \int \prod_{i=1}^{k} \Big(P(D_i|\theta_i) \prod_{j \neq i, Y_{ij}=1} P(D_j|\theta_i) \Big) \exp(-\frac{\lambda}{2} \sum_i ||\theta_i||^2) \\ & \Big(\prod_{ij} P(Y_{ij}|\vec{z}_i, \vec{z}_j, B) \prod_i P(\vec{\pi}_i|\vec{\alpha}_i) P(\vec{z}_i|\vec{\pi}_i) \Big) d(\vec{\pi}_{1:K}, \vec{Z}_{1:K}, Y) \end{aligned} \tag{37}$$

D.0.2. OPTIMIZATION

ELBO Rewrite $R_1 = (\vec{\pi}_{1:K}, \vec{Z}_{1:K}, Y)$, $R_2 = (D_{1:K})$, $R_3 = (\theta_{1:K}, \vec{\alpha}, B)$.

$$\begin{aligned} \log P(R_2, R_3) &= \log \int P(R_1, R_2, R_3) dR_1 \\ &= \log \int q(R_1) \frac{P(R_1, R_2, R_3)}{q(R_1)} dR_1 \\ &\geq \int q(R_1) \log \frac{P(R_1, R_2, R_3)}{q(R_1)} dR_1 \\ &= \mathsf{E}_q \log P(R_1, R_2, R_3) - \log q(R_1) =: \mathsf{L}(q, R_3) \end{aligned} \tag{38}$$

In E step this lower bound is maximized w.r.t q, and in M step, this lower bound is maximized w.r.t R_3. Optimal q for E step is posterior probability:

$$\begin{aligned} q^{(t)} &= P(R_1|R_2, R_3^{(t-1)}) \\ &= P(\vec{\pi}_{1:K}, \vec{Z}_{1:K}, Y|D_{1:K}, \theta_{1:K}^{(t-1)}, \vec{\alpha}^{(t-1)}, B^{(t-1)}) \end{aligned} \tag{39}$$

Mean-field approximation With mean field approximation,

$$\begin{aligned} & q_\Delta(\vec{\pi}_{1:K}, \vec{Z}_{1:K}, Y|\gamma_{1:K}, \vec{\Omega}, w) \\ =& \prod_i q_1(\pi_i|\gamma_i) q_2(\vec{z}_i|\vec{\Omega}_i, 1) \prod_{ij} q_3(Y_{ij}|w_{ij}) \end{aligned} \tag{40}$$

where q_1 is a Dirichlet, q_2 is a multinomial, q_3 is a Bernoulli, and $\Delta = (\gamma_{1:K}, \vec{\Omega}, w)$ represent the set of free variational parameters need to be estimated in the approximate distribution.

With mean-field approximation, the expectaion of the lower bound can be calculated:

$$
\begin{aligned}
&L_\Delta := \mathsf{E}_q \log P(R_1, R_2, R_3) - \log q(R_1) \\
=&E_q \Bigg[\log \prod_{i=1}^{k} \Bigg(P(D_i|\theta_i) \prod_{j \neq i, Y_{ij}=1} P(D_j|\theta_i) \Bigg) \exp(-\frac{\lambda}{2} \sum_i ||\theta_i||^2) \\
&\Bigg(\prod_{ij} P(Y_{ij}|\vec{z}_i, \vec{z}_j, B) \prod_i P(\vec{\pi}_i|\vec{\alpha}_i) P(\vec{z}_i|\vec{\pi}_i) \Bigg) - \\
&\log \prod_i q_1(\pi_i|\gamma_i) q_2(\vec{z}_i|\vec{\Omega}_i, 1) \prod_{ij} q_3(Y_{ij}|w_{ij}) \Bigg] \\
=&E_q \Bigg[\sum_{i=1}^{k} \Bigg(\log P(D_i|\theta_i) + \sum_{j \neq i} Y_{ij} \log P(D_j|\theta_i) \Bigg) - \frac{\lambda}{2} \sum_i ||\theta_i||^2 \\
&+ \sum_{i,j} Y_{ij} \log \sum_{g,h} \vec{z}_{i,g} B(g,h) \vec{z}_{j,h} + \sum_{i,j} (1 - Y_{ij}) \log(1 - \sum_{g,h} \vec{z}_{i,g} B(g,h) \vec{z}_{j,h}) \\
&+ \sum_i \sum_g \vec{z}_{i,g} \log \vec{\pi}_{i,g} \\
&+ \sum_i \sum_g (\alpha_{i,g} - 1) \log \vec{\pi}_{i,g} - \sum_i \sum_g \log \Gamma(\vec{\alpha}_{i,g}) + \sum_i \log \Gamma(\sum_g \vec{\alpha}_{i,g}) \\
&- \sum_i \sum_g (\gamma_{i,g} - 1) \log \vec{\pi}_{i,g} + \sum_i \sum_g \log \Gamma(\gamma_{i,g}) - \sum_i \log \Gamma(\sum_g \gamma_{i,g}) \\
&- \sum_i \sum_g \vec{z}_{i,g} \log \vec{\Omega}_{i,g} \\
&- \sum_{i,j} Y_{ij} \log w_{ij} - \sum_{i,j} (1 - Y_{ij}) \log(1 - w_{ij}) \Bigg] \\
=&\sum_{i=1}^{k} \Bigg(\log P(D_i|\theta_i) + \sum_{j \neq i} w_{ij} \log P(D_j|\theta_i) \Bigg) - \frac{\lambda}{2} \sum_i ||\theta_i||^2 \\
&+ \sum_{i,j} w_{ij} \sum_{g,h} \vec{\Omega}_{i,g} \vec{\Omega}_{j,h} \log B(g,h) + \sum_{i,j} (1 - w_{ij}) \sum_{g,h} \vec{\Omega}_{i,g} \vec{\Omega}_{j,h} \log(1 - B(g,h)) \\
&+ \sum_i \sum_g \vec{\Omega}_{i,g} \Bigg(\psi(\gamma_{i,g}) - \psi(\sum_k \gamma_{i,k}) \Bigg) \\
&+ \sum_i \sum_g (\vec{\alpha}_g - 1) \Bigg(\psi(\gamma_{i,g}) - \psi(\sum_k \gamma_{i,k}) \Bigg) - \sum_i \sum_g \log \Gamma(\vec{\alpha}_g) + \sum_i \log \Gamma(\sum_g \vec{\alpha}_g) \\
&- \sum_i \sum_g (\gamma_{i,g} - 1) \Bigg(\psi(\gamma_{i,g}) - \psi(\sum_k \gamma_{i,k}) \Bigg) + \sum_i \sum_g \log \Gamma(\gamma_{i,g}) - \sum_i \log \Gamma(\sum_g \gamma_{i,g}) \\
&- \sum_i \sum_g \vec{\Omega}_{i,g} \log \vec{\Omega}_{i,g} \\
&- \sum_{i,j} w_{ij} \log w_{ij} - \sum_{i,j} (1 - w_{ij}) \log(1 - w_{ij})
\end{aligned}
\tag{41}
$$

, where $\psi(x)$ is the digamma function defined as the logarithmic derivative of the gamma function: $\psi(x) = \frac{d \log \Gamma(x)}{dx}$.

E step Maximizing equation (F.0.2) w.r.t. variational parameters $\Delta = (w, \gamma_{1:K}, \vec{\Omega}_{\rightarrow}, \vec{\Omega}_{\leftarrow})$.

- w_{ij}: Setting $\nabla_w L_\Delta = 0$:

$$\begin{aligned}&\log P(D_j|\theta_i) + \sum_{g,h} \vec{\Omega}_{i,g}\vec{\Omega}_{j,h} \log B(g,h) \\ &- \sum_{g,h} \vec{\Omega}_{i,g}\vec{\Omega}_{j,h} \log(1 - B(g,h)) - 1 - \log w_{ij} + 1 + \log(1 - w_{ij}) = 0\end{aligned} \tag{42}$$

Denote

$$\begin{aligned}\hat{w_{ij}} =& \log P(D_j|\theta_i) + \sum_{g,h} \vec{\Omega}_{i,g}\vec{\Omega}_{j,h} \log B(g,h) \\ &- \sum_{g,h} \vec{\Omega}_{i,g}\vec{\Omega}_{j,h} \log(1 - B(g,h))\end{aligned} \tag{43}$$

, then

$$w^*_{ij} = \frac{1}{1 + \exp(-\hat{w_{ij}})} = \text{sigmoid}(\hat{w_{ij}}) \tag{44}$$

- $\gamma_{1:K}$: Setting $\nabla_{\gamma_{i,g}} L_\Delta = 0$:

$$\Big(\vec{\Omega}_{i,g} + \vec{\alpha}_g - \gamma_{i,g}\Big)\psi^{'}(\gamma_{i,g}) - \sum_t \Big(\vec{\Omega}_{i,t} + \vec{\alpha}_t - \gamma_{i,t}\Big)\psi^{'}(\sum_k \gamma_{i,k}) = 0 \tag{45}$$

Therefore,

$$\gamma^*_{i,g} = \vec{\Omega}_{i,g} + \vec{\alpha}_g \tag{46}$$

- $\vec{\Omega}_i$: Adding Lagrange multipliers into L_Δ, and setting $\nabla_{\vec{\Omega}_{i,k}} \Big(L_\Delta + \sum_i \lambda_i(\sum_h \vec{\Omega}_{i,h} - 1)\Big) = 0$:

$$\begin{aligned}&\sum_j w_{ij} \sum_h \vec{\Omega}_{j,h} \log B(k,h) + \sum_j w_{ji} \sum_h \vec{\Omega}_{j,h} \log B(h,k) \\ &\sum_j (1 - w_{ij}) \sum_h \vec{\Omega}_{j,h} \log(1 - B(k,h)) + \sum_j (1 - w_{ji}) \sum_h \vec{\Omega}_{j,h} \log(1 - B(h,k)) \\ &+ \psi(\gamma_{i,k}) - \psi(\sum_g \gamma_{i,g}) - 1 - \log \vec{\Omega}_{i,k} + \lambda_i = 0\end{aligned} \tag{47}$$

Therefore,

$$\begin{aligned}\vec{\Omega}^*_{i,k} \propto \exp\Bigg(&\sum_j w_{ij} \sum_h \vec{\Omega}_{j,h} \log B(k,h) + \sum_j w_{ji} \sum_h \vec{\Omega}_{j,h} \log B(h,k) \\ &\sum_j (1 - w_{ij}) \sum_h \vec{\Omega}_{j,h} \log(1 - B(k,h)) + \sum_j (1 - w_{ji}) \sum_h \vec{\Omega}_{j,h} \log(1 - B(h,k)) \\ &+ \psi(\gamma_{i,k}) - \psi(\sum_g \gamma_{i,g})\Bigg)\end{aligned} \tag{48}$$

We normalize $\vec{\Omega}^*_{i,k}$ to satisfy $\sum_k \vec{\Omega}^*_{i,k} = 1$.

M step We maximize the lower bound w.r.t. $\theta_{1:K}, \vec{\alpha}, B$.

- $\theta_{1:K}$: Using stochastic gradient ascent method,

$$\begin{aligned}\theta_i &\leftarrow \theta_i + \eta_1 \nabla_{\theta_i} L_\Delta \\ &= \theta_i - \eta_1 \Big(\nabla_{\theta_i} L(D_i;\theta_i) + \sum_{j \neq i} w_{ij} \nabla_{\theta_i} L(D_j;\theta_i) + \lambda\theta_i\Big)\end{aligned} \tag{49}$$

- $\vec{\alpha}$: Using gradient ascent method,

$$\vec{\alpha_g} \leftarrow \vec{\alpha_g} + \eta_2 \Big(\sum_i \Big(\psi(\gamma_{i,g}) - \psi(\sum_k \gamma_{i,k})\Big) - K \sum_g \psi(\vec{\alpha}_g) + K\psi(\sum_k \vec{\alpha}_k)\Big) \tag{50}$$

- B: Setting $\nabla_{B(g,h)} L_\Delta = 0$:

$$\sum_{i,j} w_{ij} \vec{\Omega}_{i,g} \vec{\Omega}_{j,h} \frac{1}{B(g,h)} - \sum_{i,j} (1 - w_{ij}) \vec{\Omega}_{i,g} \vec{\Omega}_{j,h} \frac{1}{1 - B(g,h)} = 0 \tag{51}$$

Therefore,

$$B(g,h)^* = \frac{\sum_{i,j} w_{ij} \vec{\Omega}_{i,g} \vec{\Omega}_{j,h}}{\sum_{i,j} \vec{\Omega}_{i,g} \vec{\Omega}_{j,h}} \tag{52}$$

E. SCooL-attention derivation

$$\begin{aligned}
&(Y_i \text{ is a one-hot vector for client i, drawn from categorical distribution.}) \\
&p_{ij}(\phi, \theta) = \text{softmax}(f(\theta_i; \phi)^T f(\theta_j; \phi)) \\
&\vec{Y_i} \sim \text{Categorical}(p_{i1}(\phi, \theta), p_{i2}(\phi, \theta), ..., p_{ik}(\phi, \theta)) \\
&P(Y|\theta) = \prod_{ij} p_{ij}(\phi, \theta)^{Y_{ij}} \\
&P(\theta_{1:K}) \propto \exp(-\frac{\lambda}{2} \sum_i ||\theta_i||^2) \\
&P(D_{1:K}|\theta_{1:K}, Y) = \prod_{i=1}^{K} P(D_{1:K}|\theta_i, Y) = \prod_{i=1}^{k} \left(P(D_i|\theta_i) \prod_{j \neq i, Y_{ij}=1} P(D_j|\theta_i) \right)
\end{aligned} \tag{53}$$

E.1. Objective

Our final objective is:

$$\begin{aligned}
&P(\theta_{1:K}|D_{1:K}) \propto P(\theta_{1:K}, D_{1:K}) \\
&= \int P(D_{1:K}|\theta_{1:K}, Y) P(\theta_{1:K}) P(Y|\theta_{1:K}) dY \\
&= \int \prod_{i=1}^{k} \left(P(D_i|\theta_i) \prod_{j \neq i, Y_{ij}=1} P(D_j|\theta_i) \right) \exp(-\frac{\lambda}{2} \sum_i ||\theta_i||^2) \prod_{ij} p_{ij}(\phi, \theta)^{Y_{ij}} dY
\end{aligned} \tag{54}$$

E.2. Optimization

ELBO By Jensen's inequality,

$$\begin{aligned}
\log P(\theta_{1:K}, D_{1:K}) &= \log \int P(\theta_{1:K}, D_{1:K}, Y) dY \\
&= \log \int q(Y) \frac{P(\theta_{1:K}, D_{1:K}, Y)}{q(Y)} dY \\
&\geq \int q(Y) \log \frac{P(\theta_{1:K}, D_{1:K}, Y)}{q(Y)} dY \\
&= \mathsf{E}_q \log P(\theta_{1:K}, D_{1:K}, Y)) - \log q(Y) =: \mathsf{L}_\Delta
\end{aligned} \tag{55}$$

In E step this lower bound is maximized w.r.t q, and in M step, this lower bound is maximized w.r.t R_3. Optimal q for E step is posterior probability:

$$q^{(t)} = P(Y|\theta_{1:K}^{(t-1)}) \tag{56}$$

Mean-field approximation With mean field approximation,

$$q_\Delta(Y) = \prod_i q_\Delta(\vec{Y_i}) = \text{Categorical}(p_{i1}, p_{i2}, ..., p_{iK}) \tag{57}$$

With mean-field approximation, the expectaion of the lower bound can be calculated:

$$
\begin{aligned}
&L_\Delta := \mathsf{E}_q \log P(\theta_{1:K}, D_{1:K}, Y) - \log q(Y) \\
=&E_q\Bigg[\log \prod_{i=1}^{k}\Bigg(P(D_i|\theta_i) \prod_{j\neq i, Y_{ij}=1} P(D_j|\theta_i)\Bigg) \exp(-\frac{\lambda}{2}\sum_i ||\theta_i||^2) \\
&\prod_{ij} p_{ij}(\phi,\theta)^{Y_{ij}} - \log \prod_{ij} w_{ij}^{Y_{ij}}\Bigg] \\
=&E_q\Bigg[\sum_{i=1}^{k}\Bigg(\log P(D_i|\theta_i) + \sum_{j\neq i} Y_{ij} \log P(D_j|\theta_i)\Bigg) - \frac{\lambda}{2}\sum_i ||\theta_i||^2 \\
&+ \sum_{ij} Y_{ij} \log p_{ij}(\phi,\theta) - \sum_{ij} Y_{ij} \log w_{ij}\Bigg] \\
=&\sum_{i=1}^{k}\Bigg(\log P(D_i|\theta_i) + \sum_{j\neq i} w_{ij} \log P(D_j|\theta_i)\Bigg) - \frac{\lambda}{2}\sum_i ||\theta_i||^2 \\
&+ \sum_{ij} w_{ij} \log p_{ij}(\phi,\theta) - \sum_{ij} w_{ij} \log w_{ij}
\end{aligned}
\tag{58}
$$

E step

- w_{ij}: Setting $\nabla_{w_{ij}} L_\Delta = 0$:

$$\log P(D_j|\theta_i) + \log p_{ij}(\phi,\theta) - 1 - \log w_{ij} = 0 \tag{59}$$

Then,

$$w_{ij}^* \propto \exp\Bigg(\log P(D_j|\theta_i) + \log p_{ij}(\phi,\theta)\Bigg) \tag{60}$$

We need to further normalize w_{ij} to satisfy $\sum_j w_{ij} = 1$, i.e. $w_{ij} = \frac{w_{ij}^*}{\sum_j w_{ij}^*}$.

M step

- $\theta_{1:K}$: We use stochastic gradient descent to optimize it:

$$
\begin{aligned}
&\theta_i \leftarrow \theta_i + \eta_1 \nabla_{\theta_i} L_\Delta \\
&= \theta_i - \eta_1 \Bigg(\nabla_{\theta_i} L(D_i;\theta_i) + \sum_{j\neq i} w_{ij} \nabla_{\theta_i} L(D_j;\theta_i) + \lambda\theta_i - \sum_{ij} w_{ij} \nabla_{\theta_i} \log p_{ij}(\phi,\theta) \\
&= \theta_i - \eta_1 \Bigg(\nabla_{\theta_i} L(D_i;\theta_i) + \sum_{j\neq i} w_{ij} \nabla_{\theta_i} L(D_j;\theta_i) + \lambda\theta_i \\
&+ \sum_{ij} \nabla_{\theta_i} \text{cross-entropy}(\vec{w}_i, \vec{p}_i(\phi,\theta))\Bigg)
\end{aligned}
\tag{61}
$$

- ϕ: We use stochastic gradient descent to optimize it:

$$
\begin{aligned}
&\phi \leftarrow \phi + \eta_2 \nabla_\phi L_\Delta \\
&= \phi + \eta_2 \nabla_\phi \Bigg(\sum_{ij} w_{ij} \log p_{ij}(\phi)\Bigg) \\
&= \phi - \eta_2 \sum_i \nabla_\phi \text{cross-entropy}(\vec{w}_i, \vec{p}_i(\phi,\theta))
\end{aligned}
\tag{62}
$$

F. SCooL-MMSBM

We present an additional instantiation of SCooL framework, which we use mixed membership stochastic blockmodels (MMSBM) as the prior, allowing each user to simultaneously cooperate with multiple groups of users with different

probability, thereby capturing more complex cooperation between users' tasks.

$$\begin{aligned} P(\theta_{1:K}|D_{1:K}) &\propto P(\theta_{1:K}, D_{1:K}) \\ &= \int P(D_{1:K}|\theta_{1:K}, Y)P(\theta_{1:K}|Y)P(Y)dY \end{aligned} \tag{63}$$

with

$$\begin{aligned} &P(\theta_{1:K}, Y) = P(\theta_{1:K})P(Y) \\ &P(Y) = \int P(Y, \vec{\pi}_{1:K}, Z_{\rightarrow}, Z_{\leftarrow}|\vec{\alpha}, B)d(\vec{\pi}_{1:K}, Z_{\rightarrow}, Z_{\leftarrow}) \\ &P(\theta_{1:K}) \propto \exp(-\frac{\lambda}{2}\sum_i ||\theta_i||^2) \\ &P(D_{1:K}|\theta_{1:K}, Y) = \prod_{i=1}^{K} P(D_{1:K}|\theta_i, Y) = \prod_{i=1}^{k}\Big(P(D_i|\theta_i)\prod_{j\neq i, Y_{ij}=1} P(D_j|\theta_i)\Big) \end{aligned} \tag{64}$$

F.0.1. OBJECTIVE

Modelling $P(Y)$ as MMSBM:

- For each agent $i \in K$:
  - Draw a M dimensional mixed membership vector $\vec{\pi}_i \sim$ Dirichlet$(\vec{\alpha})$.
- For each pair of agents $(i, j) \in K \times K$:
  - Draw membership indicator for the initiator, $\vec{z}_{i\rightarrow j} \sim \text{Multinomial}(\vec{\pi}_i)$.
  - Draw membership indicator for the receiver, $\vec{z}_{i\leftarrow j} \sim \text{Multinomial}(\vec{\pi}_j)$.
  - Sample the value of their interaction, $Y_{ij} \sim \text{Bernoulli}(\vec{z}_{i\rightarrow j}^{\,T} B\ \vec{z}_{i\leftarrow j})$.

Under the MMSBM, the marginal distribution of Y is:

$$\begin{aligned} P(Y|\vec{\alpha}, B) &= \int P(Y, \vec{\pi}_{1:K}, Z_{\rightarrow}, Z_{\leftarrow}|\vec{\alpha}, B)d(\vec{\pi}_{1:K}, Z_{\rightarrow}, Z_{\leftarrow}) \\ &= \int \Big(\prod_{ij} P(Y_{ij}|\vec{z}_{i\rightarrow j}, \vec{z}_{i\leftarrow j}, B)P(\vec{z}_{i\rightarrow j}|\vec{\pi}_i)P(\vec{z}_{i\leftarrow j}|\vec{\pi_j}) \\ &\qquad \prod_i P(\vec{\pi}_i|\vec{\alpha}_i)\Big)d(\vec{\pi}_{1:K}, Z_{\rightarrow}, Z_{\leftarrow}) \end{aligned} \tag{65}$$

Our final objective is:

$$\begin{aligned} &P(\theta_{1:K}|D_{1:K}) \propto P(\theta_{1:K}, D_{1:K}) \\ &= \int P(D_{1:K}|\theta_{1:K}, Y)P(\theta_{1:K})P(Y)dY \\ &= \int \prod_{i=1}^{k}\Big(P(D_i|\theta_i)\prod_{j\neq i, Y_{ij}=1} P(D_j|\theta_i)\Big)\exp(-\frac{\lambda}{2}\sum_i ||\theta_i||^2) \\ &\Big(\prod_{ij} P(Y_{ij}|\vec{z}_{i\rightarrow j}, \vec{z}_{i\leftarrow j}, B)P(\vec{z}_{i\rightarrow j}|\vec{\pi}_i)P(\vec{z}_{i\leftarrow j}|\vec{\pi_j})\prod_i P(\vec{\pi}_i|\vec{\alpha}_i)\Big)d(\vec{\pi}_{1:K}, Z_{\rightarrow}, Z_{\leftarrow}, Y) \end{aligned} \tag{66}$$

F.0.2. OPTIMIZATION

ELBO Rewrite $R_1 = (Y, \vec{\pi}_{1:K}, Z_{\rightarrow}, Z_{\leftarrow})$, $R_2 = (D_{1:K})$, $R_3 = (\theta_{1:K}, \vec{\alpha}, B)$.

$$\begin{aligned}
\log P(R_2, R_3) &= \log \int P(R_1, R_2, R_3) dR_1 \\
&= \log \int q(R_1) \frac{P(R_1, R_2, R_3)}{q(R_1)} dR_1 \\
&\geq \int q(R_1) \log \frac{P(R_1, R_2, R_3)}{q(R_1)} dR_1 \\
&= \mathsf{E}_q \log P(R_1, R_2, R_3) - \log q(R_1) =: \mathsf{L}(q, R_3)
\end{aligned} \tag{67}$$

In E step this lower bound is maximized w.r.t q, and in M step, this lower bound is maximized w.r.t R_3. Optimal q for E step is posterior probability:

$$\begin{aligned}
q^{(t)} &= P(R_1 | R_2, R_3^{(t-1)}) \\
&= P(Y, \vec{\pi}_{1:K}, Z_{\rightarrow}, Z_{\leftarrow} | D_{1:K}, \theta_{1:K}^{(t-1)}, \vec{\alpha}^{(t-1)}, B^{(t-1)})
\end{aligned} \tag{68}$$

Mean-field approximation With mean field approximation,

$$\begin{aligned}
&q_{\Delta}(Y, \vec{\pi}_{1:K}, Z_{\rightarrow}, Z_{\leftarrow} | w, \gamma_{1:K}, \vec{\phi}_{\rightarrow}, \vec{\phi}_{\leftarrow}) \\
=&\prod_i q_1(\pi_i | \gamma_i) \prod_{i,j} q_2(\vec{z}_{i \rightarrow j} | \vec{\phi}_{i \rightarrow j}, 1) q_2(\vec{z}_{i \leftarrow j} | \vec{\phi}_{i \leftarrow j}, 1) \prod_{ij} q_3(Y_{ij} | w_{ij})
\end{aligned} \tag{69}$$

where q_1 is a Dirichlet, q_2 is a multinomial, q_3 is a Bernoulli, and $\Delta = (w, \gamma_{1:K}, \vec{\phi}_{\rightarrow}, \vec{\phi}_{\leftarrow})$ represent the set of free variational parameters need to be estimated in the approximate distribution.

With mean-field approximation, the expectaion of the lower bound can be calculated:

$$\begin{aligned}
&L_{\Delta} := \mathsf{E}_q \log P(R_1, R_2, R_3) - \log q(R_1) \\
=&E_q \Bigg[\log \prod_{i=1}^{k} \Bigg(P(D_i | \theta_i) \prod_{j \neq i, Y_{ij}=1} P(D_j | \theta_i) \Bigg) \exp(-\frac{\lambda}{2} \sum_i ||\theta_i||^2) \\
&\Bigg(\prod_{ij} P(Y_{ij} | \vec{z}_{i \rightarrow j}, \vec{z}_{i \leftarrow j}, B) P(\vec{z}_{i \rightarrow j} | \vec{\pi}_i) P(\vec{z}_{i \leftarrow j} | \vec{\pi_j}) \prod_i P(\vec{\pi}_i | \vec{\alpha}_i) \Bigg) -
\end{aligned}$$

$$
\begin{aligned}
&\log \prod_i q_1(\pi_i|\gamma_i) \prod_{i,j} q_2(\vec{z}_{i\to j}|\vec{\phi}_{i\to j},1) q_2(\vec{z}_{i\leftarrow j}|\vec{\phi}_{i\leftarrow j},1) \prod_{ij} q_3(Y_{ij}|w_{ij}) \Bigg] \\
=&E_q \Bigg[\sum_{i=1}^{k} \Bigg(\log P(D_i|\theta_i) + \sum_{j\neq i} Y_{ij} \log P(D_j|\theta_i) \Bigg) - \frac{\lambda}{2} \sum_i ||\theta_i||^2 \\
&+ \sum_{i,j} Y_{ij} \log \sum_{g,h} \vec{z}_{i\to j,g} B(g,h) \vec{z}_{i\leftarrow j,h} + \sum_{i,j} (1-Y_{ij}) \log(1 - \sum_{g,h} \vec{z}_{i\to j,g} B(g,h) \vec{z}_{i\leftarrow j,h}) \\
&+ \sum_{i,j} \sum_g \vec{z}_{i\to j,g} \log \vec{\pi}_{i,g} + \sum_{i,j} \sum_g \vec{z}_{i\leftarrow j,g} \log \vec{\pi}_{j,g} \\
&+ \sum_i \sum_g (\alpha_{i,g} - 1) \log \vec{\pi}_{i,g} - \sum_i \sum_g \log \Gamma(\vec{\alpha}_{i,g}) + \sum_i \log \Gamma(\sum_g \vec{\alpha}_{i,g}) \\
&- \sum_i \sum_g (\gamma_{i,g} - 1) \log \vec{\pi}_{i,g} + \sum_i \sum_g \log \Gamma(\gamma_{i,g}) - \sum_i \log \Gamma(\sum_g \gamma_{i,g}) \\
&- \sum_{i,j} \sum_g \vec{z}_{i\to j,g} \log \vec{\phi}_{i\to j,g} - \sum_{i,j} \sum_g \vec{z}_{i\leftarrow j,g} \log \vec{\phi}_{i\leftarrow j,g} \\
&- \sum_{i,j} Y_{ij} \log w_{ij} - \sum_{i,j} (1-Y_{ij}) \log(1-w_{ij}) \Bigg] \\
=&\sum_{i=1}^{k} \Bigg(\log P(D_i|\theta_i) + \sum_{j\neq i} w_{ij} \log P(D_j|\theta_i) \Bigg) - \frac{\lambda}{2} \sum_i ||\theta_i||^2 \\
&+ \sum_{i,j} w_{ij} \sum_{g,h} \vec{\phi}_{i\to j,g} \vec{\phi}_{i\leftarrow j,h} \log B(g,h) + \sum_{i,j} (1-w_{ij}) \sum_{g,h} \vec{\phi}_{i\to j,g} \vec{\phi}_{i\leftarrow j,h} \log(1 - B(g,h)) \\
&+ \sum_{i,j} \sum_g \vec{\phi}_{i\to j,g} \Bigg(\psi(\gamma_{i,g}) - \psi(\sum_k \gamma_{i,k}) \Bigg) + \sum_{i,j} \sum_g \vec{\phi}_{i\leftarrow j,g} \Bigg(\psi(\gamma_{j,g}) - \psi(\sum_k \gamma_{j,k}) \Bigg) \\
&+ \sum_i \sum_g (\vec{\alpha}_g - 1) \Bigg(\psi(\gamma_{i,g}) - \psi(\sum_k \gamma_{i,k}) \Bigg) - \sum_i \sum_g \log \Gamma(\vec{\alpha}_g) + \sum_i \log \Gamma(\sum_g \vec{\alpha}_g) \\
&- \sum_i \sum_g (\gamma_{i,g} - 1) \Bigg(\psi(\gamma_{i,g}) - \psi(\sum_k \gamma_{i,k}) \Bigg) + \sum_i \sum_g \log \Gamma(\gamma_{i,g}) - \sum_i \log \Gamma(\sum_g \gamma_{i,g}) \\
&- \sum_{i,j} \sum_g \vec{\phi}_{i\to j,g} \log \vec{\phi}_{i\to j,g} - \sum_{i,j} \sum_g \vec{\phi}_{i\leftarrow j,g} \log \vec{\phi}_{i\leftarrow j,g} \\
&- \sum_{i,j} w_{ij} \log w_{ij} - \sum_{i,j} (1-w_{ij}) \log(1-w_{ij})
\end{aligned}
$$

, where $\psi(x)$ is the digamma function defined as the logarithmic derivative of the gamma function: $\psi(x) = \frac{d \log \Gamma(x)}{dx}$.

E step Maximizing equation F.0.2 w.r.t. variational parameters $\Delta = (w, \gamma_{1:K}, \vec{\phi}_{\to}, \vec{\phi}_{\leftarrow})$.

- w_{ij}: Setting $\nabla_w L_\Delta = 0$:

$$
\begin{aligned}
&\log P(D_j|\theta_i) + \sum_{g,h} \vec{\phi}_{i\to j,g} \vec{\phi}_{i\leftarrow j,h} \log B(g,h) \\
&- \sum_{g,h} \vec{\phi}_{i\to j,g} \vec{\phi}_{i\leftarrow j,h} \log(1 - B(g,h)) - 1 - \log w_{ij} + 1 + \log(1-w_{ij}) = 0
\end{aligned} \tag{70}
$$

Denote

$$\begin{aligned}\hat{w_{ij}} =& \log P(D_j|\theta_i) + \sum_{g,h} \vec{\phi}_{i\to j,g}\vec{\phi}_{i\leftarrow j,h} \log B(g,h) \\ &- \sum_{g,h} \vec{\phi}_{i\to j,g}\vec{\phi}_{i\leftarrow j,h} \log(1-B(g,h))\end{aligned} \tag{71}$$

, then

$$w^*_{ij} = \frac{1}{1+\exp(-\hat{w_{ij}})} = \text{sigmoid}(\hat{w_{ij}}) \tag{72}$$

- $\gamma_{1:K}$: Setting $\nabla_{\gamma_{i,g}} L_\Delta = 0$:

$$\begin{aligned}&\sum_j \vec{\phi}_{i\to j,g}\Big(\psi^{'}(\gamma_{i,g}) - \psi^{'}(\sum_k \gamma_{i,k})\Big) + \sum_j \vec{\phi}_{j\leftarrow i,g}\Big(\psi^{'}(\gamma_{i,g}) - \psi^{'}(\sum_k \gamma_{i,k})\Big) \\ &+ (\vec{\alpha}_g - 1)\Big(\psi^{'}(\gamma_{i,g}) - \psi^{'}(\sum_k \gamma_{i,k})\Big) - \Big(\psi(\gamma_{i,g}) - \psi(\sum_k \gamma_{i,k})\Big) \\ &- (\gamma_{i,g}-1)\Big(\psi^{'}(\gamma_{i,g}) - \psi^{'}(\sum_k \gamma_{i,k})\Big) + \psi(\gamma_{i,g}) - \psi(\sum_k \gamma_{i,k}) = 0\end{aligned} \tag{73}$$

$$\Big(\sum_j \vec{\phi}_{i\to j,g} + \sum_j \vec{\phi}_{j\leftarrow i,g} + \vec{\alpha}_g - \gamma_{i,g}\Big)\Big(\psi^{'}(\gamma_{i,g}) - \psi^{'}(\sum_k \gamma_{i,k})\Big) = 0$$

Therefore,

$$\gamma^*_{i,g} = \vec{\alpha}_g + \sum_j \vec{\phi}_{i\to j,g} + \sum_j \vec{\phi}_{j\leftarrow i,g} \tag{74}$$

- $\vec{\phi}_{i\to j}$: Adding Lagrange multipliers into L_Δ, and setting $\nabla_{\vec{\phi}_{i\to j,k}}\Big(L_\Delta + \sum_{i,j}\lambda_{ij}(\sum_h \vec{\phi}_{i\to j,h} - 1)\Big) = 0$:

$$\begin{aligned}&w_{ij}\sum_h \vec{\phi}_{i\leftarrow j,h}\log B(g,h) + (1-w_{ij})\sum_h \vec{\phi}_{i\leftarrow j,h}\log(1-B(g,h)) \\ &+ \psi(\gamma_{i,k}) - \psi(\sum_g \gamma_{i,g}) - 1 - \log\vec{\phi}_{i\to j,k} + \lambda_{ij} = 0\end{aligned} \tag{75}$$

Therefore,

$$\begin{aligned}\vec{\phi}^*_{i\to j,k} \propto \exp\Big(&w_{ij}\sum_h \vec{\phi}_{i\leftarrow j,h}\log B(k,h) + \\ &(1-w_{ij})\sum_h \vec{\phi}_{i\leftarrow j,h}\log(1-B(k,h)) + \psi(\gamma_{i,k}) - \psi(\sum_g \gamma_{i,g})\Big)\end{aligned} \tag{76}$$

We normalize $\vec{\phi}^*_{i\to j,k}$ to satisfy $\sum_k \vec{\phi}^*_{i\to j,k} = 1$.

- $\vec{\phi}_{i\leftarrow j}$: Following similar derivations as $\vec{\phi}_{i\to j}$, we get:

$$\begin{aligned}\vec{\phi}^*_{i\leftarrow j,h} \propto \exp\Big(&w_{ij}\sum_g \vec{\phi}_{i\to j,g}\log B(g,h) + \\ &(1-w_{ij})\sum_g \vec{\phi}_{i\to j,g}\log(1-B(g,h)) + \psi(\gamma_{j,h}) - \psi(\sum_g \gamma_{j,g})\Big)\end{aligned} \tag{77}$$

We normalize $\vec{\phi}^*_{i\leftarrow j,h}$ to satisfy $\sum_h \vec{\phi}^*_{i\leftarrow j,h} = 1$.

M step We maximize the lower bound w.r.t. $\theta_{1:K}, \vec{\alpha}, B$.

- $\theta_{1:K}$: Using stochastic gradient ascent method,

$$\begin{aligned}\theta_i &\leftarrow \theta_i + \eta_1 \nabla_{\theta_i} L_\Delta \\ &= \theta_i - \eta_1 \bigg(\nabla_{\theta_i} L(D_i; \theta_i) + \sum_{j \neq i} w_{ij} \nabla_{\theta_i} L(D_j; \theta_i) + \lambda \theta_i \bigg)\end{aligned} \tag{78}$$

- $\vec{\alpha}$: Using gradient ascent method,

$$\vec{\alpha_g} \leftarrow \vec{\alpha_g} + \eta_2 \bigg(\sum_i \bigg(\psi(\gamma_{i,g}) - \psi(\sum_k \gamma_{i,k}) \bigg) - K \sum_g \psi(\vec{\alpha}_g) + K \psi(\sum_k \vec{\alpha}_k) \bigg) \tag{79}$$

- B: Setting $\nabla_{B(g,h)} L_\Delta = 0$:

$$\sum_{i,j} w_{ij} \vec{\phi}_{i \to j,g} \vec{\phi}_{i \leftarrow j,h} \frac{1}{B(g,h)} - \sum_{i,j} (1 - w_{ij}) \vec{\phi}_{i \to j,g} \vec{\phi}_{i \leftarrow j,h} \frac{1}{1 - B(g,h)} = 0 \tag{80}$$

Therefore,

$$B(g,h)^* = \frac{\sum_{i,j} w_{ij} \vec{\phi}_{i \to j,g} \vec{\phi}_{i \leftarrow j,h}}{\sum_{i,j} \vec{\phi}_{i \to j,g} \vec{\phi}_{i \leftarrow j,h}} \tag{81}$$