PERSONALIZED DECENTRALIZED BILEVEL OPTIMIZATION OVER STOCHASTIC AND DIRECTED NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

While personalization in distributed learning has been extensively studied, existing approaches employ dedicated algorithms to optimize their specific type of parameters (e.g., client clusters or model interpolation weights), making it difficult to simultaneously optimize different types of parameters to yield better performance. Moreover, their algorithms require centralized or static undirected communication networks, which can be vulnerable to center-point failures or deadlocks. This study proposes optimizing various types of parameters using a single algorithm that runs on more practical communication environments. First, we propose a gradient-based bilevel optimization that reduces most personalization approaches to the optimization of client-wise hyperparameters. Second, we propose a decentralized algorithm to estimate gradients with respect to the hyperparameters, which can run even on stochastic and directed communication networks. Our empirical results demonstrated that the gradient-based bilevel optimization enabled combining existing personalization approaches which led to state-of-the-art performance, confirming it can perform on multiple simulated communication environments including a stochastic and directed network.

1 INTRODUCTION

In distributed learning, providing personally tuned models to clients, or personalization, has shown to be effective when the clients' data are heterogeneously distributed (Tan et al., 2022).

While various approaches have been proposed, they are dedicated to optimizing specific types of parameters for personalization. A typical example is clustering-based personalization (Sattler et al., 2020), which employs similarity-based clustering specifically for seeking client clusters. Another approach called model interpolation (Mansour et al., 2020; Deng et al., 2020) also specializes in optimizing interpolation weights between local and global models. These dedicated algorithms prevent developers from combining different personalization methods to achieve better performance.

Another limitation of previous personalization algorithms is that they can run only on centralized or static undirected networks. Most approaches for federated learning (Smith et al., 2017; Sattler et al., 2020; Jiang et al., 2019) require centralized settings in which a host server can communicate with any client. Although a few studies (Lu et al., 2022; Marfoq et al., 2021) consider fully-decentralized settings, they assume that the communication edge between any clients is static and undirected (i.e., synchronized). These commutation networks are known to be vulnerable to practical issues, such as bottlenecks or central point failures on the host servers (Assran et al., 2019), or failing nodes and deadlocks on the static undirected networks (Tsianos et al., 2012).

This study proposes optimizing various parameters for personalization using a single algorithm while allowing more practical communication environments. First, we propose a gradient-based *Personalized Decentralized Bilevel Optimization (PDBO)*, which reduces many personalization approaches to the optimization of hyperparameters possessed by each client. Second, we propose *Hyper-gradient Push (HGP)* that allows any client to solve PDBO by estimating the gradient with respect to its hyperparameters (*hyper-gradient*) via stochastic and directed communications, that are immune to the practical problems of centralized or static undirected communications (Assran et al., 2019). We also introduce a variance-reduced HGP to avoid estimation variance, which is particularly effective when communications are stochastic, providing its theoretical error bound.

We empirically demonstrated that the generality of our gradient-based PDBO enabled combining existing personalization approaches which led to state-of-the-art performance in a distributed classification task. We also demonstrated that the gradient-based PDBO succeeded in the personalization on multiple simulated communication environments including a stochastic and directed network.

Our contributions are summarized as follows:

- We propose a gradient-based PDBO that can solve existing personalization problems and their combinations as its special cases.
- We propose a decentralized hyper-gradient estimation algorithm called HGP which can run even on stochastic and directed networks. We also propose a variance-reduced HGP, which is particularly effective in stochastic communications, and provide its theoretical error bound.
- We empirically validated the advantages of the gradient-based PDBO with HGP; it enabled solving a combination of different personalization problems which led to state-of-the-art performance, and it performed on different communication environments including a stochastic directed network.

Notation $\langle \boldsymbol{A} \rangle_{ij}$ denotes the matrix at the i-th row and j-th column block of the matrix $\boldsymbol{A}$, and $\langle \boldsymbol{a} \rangle_i$ denotes the i-th block vector of the vector $\boldsymbol{a}$. For a function $f : \mathbb{R}^{d_1} \mapsto \mathbb{R}^{d_2}$, we denote its total and partial derivatives with respect to a vector $\boldsymbol{x} \in \mathbb{R}^{d_1}$ by $\mathrm{d}_{\boldsymbol{x}} f(\boldsymbol{x}) \in \mathbb{R}^{d_1 \times d_2}$ and $\partial_{\boldsymbol{x}} f(\boldsymbol{x}) \in \mathbb{R}^{d_1 \times d_2}$, respectively. We denote the product of matrices by $\prod_{s=0}^{m} \hat{\boldsymbol{A}}^{(s)} = \hat{\boldsymbol{A}}^{(m)} \cdots \hat{\boldsymbol{A}}^{(0)}$ and $\prod_{s=0}^{-1} \hat{\boldsymbol{A}}^{(s)} = \boldsymbol{I}$.

2 PRELIMINARIES

We formulate distributed learning, communication networks, and stochastic gradient push (Nedić & Olshevsky, 2016, SGP) as a generalization of current gradient-based distributed learning.

Distributed learning Distributed learning (Li et al., 2014; McMahan et al., 2017) with n clients is commonly formulated for all $i \in [n]$ as follows:

$$\boldsymbol{x}_i^* = \arg\min_{\boldsymbol{x}_i} \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\xi_i} \left[f_i \left(\boldsymbol{x}_i, \boldsymbol{\lambda}_i; \xi_i \right) \right], \quad \text{s.t. } \boldsymbol{x}_i = \boldsymbol{x}_j, \forall j \in [n], \tag{1}$$

where, the i-th client pursues the optimal parameter $\boldsymbol{x}_i^* \in \mathbb{R}^{d_{\boldsymbol{x}}}$, that makes *consensus* ($\boldsymbol{x}_i = \boldsymbol{x}_j, \forall j \in [n]$) over all the clients, while minimizing its cost $f_i : \mathbb{R}^{d_{\boldsymbol{x}}} \times \mathbb{R}^{d_{\boldsymbol{\lambda}}} \mapsto \mathbb{R}$ for the input $\xi_i \in \mathcal{X}$ sampled from its local distribution $p(\xi_i)$. We allow f_i to take the hyperparameters $\boldsymbol{\lambda}_i \in \mathbb{R}^{d_{\boldsymbol{\lambda}}}$ as its argument. We further explain the examples of the choice of $\boldsymbol{\lambda}_i$ in Sections 3 and 5.

Stochastic and directed communication network In distributed learning, clients solve Eq. (1) by exchanging messages over a physical communication network. The type of edge connections categorizes the communication network: *static undirected* (Lian et al., 2017), which represents synchronization over all clients; *stochastic undirected* (Lian et al., 2018), which represents asynchronicity between different client pairs; and *stochastic directed* (Nedić & Olshevsky, 2016), which represents push communication where any message passing can be unidirectional.

This study considers distributed learning on stochastic and directed communication networks. Such a network has several desirable properties: robustness to failing clients and deadlocks (Tsianos et al., 2012), immunity to central failures, and small communication overhead (Assran et al., 2019). We model stochastic directed networks by letting communication edges be randomly realized, as simulated in Assran et al. (2019) and Nedić & Olshevsky (2016). Let $\delta_{i \to j}^{(t)} \in \{0, 1\}$ be a random variable where $\delta_{i \to j}^{(t)} = 1$ denotes that there is a communication channel from the i-th client to the j-th client at the time step t, and $\delta_{i \to j}^{(t)} = 0$ otherwise. We set $\delta_{i \to i}^{(t)} = 1$ for all $i \in [n]$ and $t \in \mathbb{N}$ allowing every client to send a message to itself at any time step. Note that the edge model above can recover the other fully-decentralized settings as its special cases; the symmetric edges ($\delta_{i \to j}^{(t)} = \delta_{j \to i}^{(t)}, \forall i, j \in [n], \forall t \in \mathbb{N}$) recover stochastic undirected networks, and the symmetric constant edges, which additionally require $\delta_{i \to j}^{(t)} = \delta_{j \to i}^{(t)} = \delta_{ij}$, recover static and undirected networks.

Stochastic gradient push (SGP) SGP (Nedić & Olshevsky, 2016; Assran et al., 2019) is one of the most general methods for solving the problem Eq. (1). This section briefly describes SGP with further generalizations for its variants.

The i-th client in SGP updates its weight $\omega_i \in \mathbb{R}$ along with biased parameter $\boldsymbol{z}_i \in \mathbb{R}^{d_{\boldsymbol{x}}}$ to obtain its debiased parameter $\boldsymbol{x}_i = \boldsymbol{z}_i/\omega_i$. Let $\boldsymbol{y}_i = [\boldsymbol{z}_i^\top \;\; \omega_i]^\top \in \mathbb{R}^{d_{\boldsymbol{y}}}$. At the t-th step, the i-th client samples its minibatch $\zeta_i^{(t)}$ and sending edges $\delta_i^{(t)} = [\delta_{i\to 1}^{(t)} \;\cdots\; \delta_{i\to n}^{(t)}]^\top$, and runs a local update $\boldsymbol{\psi}_i : \mathbb{R}^{d_{\boldsymbol{y}}} \mapsto \mathbb{R}^{d_{\boldsymbol{y}}}$ and message generator $\boldsymbol{\varphi}_i : \mathbb{R}^{d_{\boldsymbol{y}}} \mapsto \mathbb{R}^{d_{\boldsymbol{y}}}$ for whom the i-th client sends to (i.e., $\delta_{i\to j}^{(t)} = 1$). Then, the i-th client update $\boldsymbol{y}_i$ as

$$\boldsymbol{y}_i^{(t+1)} = \sum_{j=1}^{n} p_{ji}(\delta_j^{(t)})\boldsymbol{\varphi}_j\left(\boldsymbol{y}_j^{(t)}; \boldsymbol{\lambda}_j, \zeta_j^{(t)}\right) + \boldsymbol{\psi}_i\left(\boldsymbol{y}_i^{(t)}; \boldsymbol{\lambda}_i, \zeta_i^{(t)}\right), \tag{2}$$

$$\text{s.t.} \quad \sum_{k=1}^{n} p_{ik}\left(\delta_i^{(t)}\right) = 1 \quad \text{and} \quad p_{ik}\left(\delta_i^{(t)}\right) = \delta_{i\to k}^{(t)} p_{ik}\left(\delta_i^{(t)}\right), \forall k \in [n], \tag{3}$$

where, $p_{ij} : \{0,1\}^n \mapsto [0,1]$ ensures the convergence of $\boldsymbol{x}_i$ to the consensus. Denoting the learning rate by $\alpha_i \in \mathbb{R}^+$, the following formulations of $\boldsymbol{\varphi}_i$ and $\boldsymbol{\psi}_i$ recover the two SGP variants:

$$\begin{cases} \boldsymbol{\varphi}_i\left(\boldsymbol{y}_i; \boldsymbol{\lambda}_i, \zeta_i\right) = \left[\boldsymbol{z}_i^\top - \dfrac{\alpha_i}{|\zeta_i|}\displaystyle\sum_{\xi\in\zeta_i} \partial_{\boldsymbol{x}_i} f_i\left(\dfrac{\boldsymbol{z}_i}{\omega_i}, \boldsymbol{\lambda}_i; \xi\right)^\top \;\; w_i\right]^\top, \;\; \boldsymbol{\psi}_i\left(\boldsymbol{y}_i; \boldsymbol{\lambda}_i, \zeta_i\right) = \left[\boldsymbol{0}_{d_{\boldsymbol{x}}} \;\; 0\right]^\top & \text{(4a)} \\ \boldsymbol{\varphi}_i\left(\boldsymbol{y}_i; \boldsymbol{\lambda}_i, \zeta_i\right) = \left[\boldsymbol{z}_i^\top \;\; w_i\right]^\top, \;\; \boldsymbol{\psi}_i\left(\boldsymbol{x}_i; \boldsymbol{\lambda}_i, \zeta_i\right) = \left[-\dfrac{\alpha_i}{|\zeta_i|}\displaystyle\sum_{\xi\in\zeta_i} \partial_{\boldsymbol{x}_i} f_i\left(\dfrac{\boldsymbol{z}_i}{\omega_i}, \boldsymbol{\lambda}_i; \xi\right)^\top \;\; 0\right]^\top, & \text{(4b)} \end{cases}$$

where Eq. (4a) and Eq. (4b) run local gradient descent with a minibatch before (Assran et al., 2019) and after (Nedić & Olshevsky, 2016) communication, respectively.

We can recover other popular distributed learning schemes as special cases of SGP. By making p_{ij} form a doubly stochastic mixing matrix, Eq. (4a) and Eq. (4b) recover the decentralized stochastic gradient descent (DSGD) in Bianchi et al. (2013) and Lian et al. (2017), respectively. We can also recover FedAVG (McMahan et al., 2017) by choosing a fully-connected graph with averaging over all clients, i.e., $\delta_{i\to j}^{(t)} = 1$ and $p_{ij}(\delta_i^{(t)}) = 1/n$ for all $i, j \in [n]$ and $t \in \mathbb{N}$ in Eq. (4a).[1]

3 Personalized Decentralized Bilevel Optimization (PDBO)

We then propose the formulation of PDBO as a generalization of existing personalization problems. PDBO played by n clients is formulated as follows for all $i \in [n]$:

$$\min_{\boldsymbol{\lambda}_i} \frac{1}{n}\sum_{i=1}^{n} F_i\left(\boldsymbol{x}_i^*\left(\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_n\right), \boldsymbol{\lambda}_i\right), \text{ s.t. Eq. (1)}, \tag{5}$$

where the outer-problem (Eq. (5-left)) lets the i-th client find its optimal hyperparameter $\boldsymbol{\lambda}_i$ that minimizes the average of outer-cost $F_i : \mathbb{R}^{d_{\boldsymbol{x}}} \times \mathbb{R}^{d_{\boldsymbol{\lambda}}} \mapsto \mathbb{R}$ across all clients. Here, we write $\boldsymbol{x}_i^*\left(\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_n\right)$ to show its dependency to hyper-parameters explicitly. The generality of Eq. (5) in personalization comes from the flexibility in the choice of $f_i, F_i, \boldsymbol{x}_i$, and $\boldsymbol{\lambda}_i$. For example, suppose that f_i is the cross-entropy loss of a DNN with a feature extractor and classifier parameterized by $\boldsymbol{x}_i$ and $\boldsymbol{\lambda}_i$, respectively. By letting F_i be a validation loss, we can recover a family of personalized layer scheme (Arivazhagan et al., 2019; Bui et al., 2019). See Section 5 for further examples.

We then reformulate PDBO assuming that Eq. (1) is solved by T steps of the SGP update (Eq. (2)):

$$\min_{\boldsymbol{\lambda}_i} \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[F_i\left(\boldsymbol{x}_i^{(T)}\left(\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_n\right), \boldsymbol{\lambda}_i\right)\right], \text{ s.t. Eq. (2) for } t = 0, 1, \ldots, T-1, \tag{6}$$

where, the expectation is taken over by the randomness in the selection of minibatch $\zeta_i^{(t)}$ and the realization of communication edges $\delta_{i\to j}^{(t)}$ for all $i, j \in [n]$ and $t = 0, 1, \ldots, T-1$. This reformulation barely loses the generality of Eq. (5) because of the generality of Eq. (2) presented in Section 2.

[1]This expression is for convention to show the mathematical equivalence. In practice, FedAVG runs on a centralized network with a central server.

4 HYPER-GRADIENT ESTIMATION OVER STOCHASTIC AND DIRECTED COMMUNICATION NETWORKS

To solve PDBO using gradient-based methods, this section introduces an empirical estimate of the hyper-gradient and its decentralized computation algorithm, which we named HGP.

4.1 EMPIRICAL ESTIMATE VIA APPROXIMATE IMPLICIT DIFFERENTIATION

Let $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{x}_1^\top & \cdots & \boldsymbol{x}_n^\top\end{bmatrix}^\top$, $\boldsymbol{\lambda} = \begin{bmatrix}\boldsymbol{\lambda}_1^\top & \cdots & \boldsymbol{\lambda}_n^\top\end{bmatrix}^\top$, and $\bar{F}(\boldsymbol{x}, \boldsymbol{\lambda}) := \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[F_i\left(\boldsymbol{x}_i, \boldsymbol{\lambda}_i\right)\right]$. The expected hyper-gradient with respect to $\boldsymbol{\lambda}$ is written as $\mathrm{d}_{\boldsymbol{\lambda}}\bar{F}(\boldsymbol{x}^{(T)}, \boldsymbol{\lambda})$ under Assumption 1.

Assumption 1. For all $i \in [n]$, $\boldsymbol{\varphi}_i\left(\boldsymbol{y}_i; \boldsymbol{\lambda}_i, \zeta_i\right)$ and $\boldsymbol{\psi}_i\left(\boldsymbol{y}_i; \boldsymbol{\lambda}_i, \zeta_i\right)$ are differentiable with respect to $\boldsymbol{y}_i$ and $\boldsymbol{\lambda}_i$, and $F_i\left(\boldsymbol{x}_i, \boldsymbol{\lambda}_i\right)$ is differentiable with respect to $\boldsymbol{x}_i$ and $\boldsymbol{\lambda}_i$.

Below, we derive the estimator of $\mathrm{d}_{\boldsymbol{\lambda}}\bar{F}(\boldsymbol{x}^{(T)}, \boldsymbol{\lambda})$ following the recurrent backpropagation for approximate implicit differentiation (Grazzi et al., 2020).

Approximate implicit differentiation To apply implicit differentiation, we introduce the following assumption as in Grazzi et al. (2020).

Assumption 2. Let $\delta = \{\delta_1, \ldots, \delta_n\}$ and $\zeta = \{\zeta_1, \ldots, \zeta_n\}$. $\boldsymbol{y}_i^* = \boldsymbol{y}_i^{(T)}$ is the unique stationary point that satisfies $\boldsymbol{y}_i^* = [\boldsymbol{z}_i^{*\top} \;\; \omega_i^*]^\top = \mathbb{E}_{\delta,\zeta}\left[\sum_{j=1}^n p_{ji}\left(\delta_j\right)\boldsymbol{\varphi}_j(\boldsymbol{y}_j^*; \boldsymbol{\lambda}_j, \zeta_j) + \boldsymbol{\psi}_i(\boldsymbol{y}_i^*; \boldsymbol{\lambda}_i, \zeta_i)\right]$, $\forall i \in [n]$.

Let $\boldsymbol{y}^*\left(\boldsymbol{\lambda}\right) = \begin{bmatrix}\boldsymbol{y}_1^{*\top} & \cdots & \boldsymbol{y}_n^{*\top}\end{bmatrix}^\top$. We introduce Jacobian matrices $\boldsymbol{A}\left(\delta, \zeta\right)$ and $\boldsymbol{B}\left(\delta, \zeta\right)$ whose (j, i) blocks are the partial derivative of Eq. (2) with respect to $\boldsymbol{y}_j$ and $\boldsymbol{\lambda}_j$ for $j, i \in [n]$, respectively:

$$\left\langle \boldsymbol{A}\left(\delta, \zeta\right)\right\rangle_{ji} = p_{ji}\left(\delta_j\right)\partial_{\boldsymbol{y}_j}\boldsymbol{\varphi}_j\left(\boldsymbol{y}_j^*; \boldsymbol{\lambda}_j, \zeta_j\right) + \mathbb{1}_{ji}\partial_{\boldsymbol{y}_j}\boldsymbol{\psi}_j\left(\boldsymbol{y}_j^*; \boldsymbol{\lambda}_j, \zeta_j\right) \in \mathbb{R}^{d_{\boldsymbol{y}} \times d_{\boldsymbol{y}}}, \tag{7a}$$

$$\left\langle \boldsymbol{B}\left(\delta, \zeta\right)\right\rangle_{ji} = p_{ji}\left(\delta_j\right)\partial_{\boldsymbol{\lambda}_j}\boldsymbol{\varphi}_j\left(\boldsymbol{y}_j^*; \boldsymbol{\lambda}_j, \zeta_j\right) + \mathbb{1}_{ji}\partial_{\boldsymbol{\lambda}_j}\boldsymbol{\psi}_j\left(\boldsymbol{y}_j^*; \boldsymbol{\lambda}_j, \zeta_j\right) \in \mathbb{R}^{d_{\boldsymbol{\lambda}} \times d_{\boldsymbol{y}}}, \tag{7b}$$

where, $\mathbb{1}_{ij}$ denotes the Kronecker delta. We introduce their expectations by $\bar{\boldsymbol{A}} := \mathbb{E}_{\delta,\zeta}[\boldsymbol{A}\left(\delta, \zeta\right)]$ and $\bar{\boldsymbol{B}} := \mathbb{E}_{\delta,\zeta}[\boldsymbol{B}\left(\delta, \zeta\right)]$ assuming the following:

Assumption 3. The largest singular value of $\bar{\boldsymbol{A}}$ is strictly smaller than one.

Using Assumption 2, we can express the stationarity condition of Eq. (2) as

$$\mathrm{d}_{\boldsymbol{\lambda}}\boldsymbol{y}^*\left(\boldsymbol{\lambda}\right) = \mathrm{d}_{\boldsymbol{\lambda}}\boldsymbol{y}^*\left(\boldsymbol{\lambda}\right)\bar{\boldsymbol{A}} + \bar{\boldsymbol{B}} \quad \Rightarrow \quad \mathrm{d}_{\boldsymbol{\lambda}}\boldsymbol{y}^*\left(\boldsymbol{\lambda}\right) = \bar{\boldsymbol{B}}(\boldsymbol{I} - \bar{\boldsymbol{A}})^{-1},$$

Assumption 3 ensures the existence of the inverse $(\boldsymbol{I} - \bar{\boldsymbol{A}})^{-1}$ and allows truncated Neumann series approximation $(\boldsymbol{I} - \bar{\boldsymbol{A}})^{-1} = \sum_{m=0}^{\infty}\bar{\boldsymbol{A}}^m \approx \sum_{m=0}^{M-1}\bar{\boldsymbol{A}}^m$, which leads to

$$\mathrm{d}_{\boldsymbol{\lambda}}\bar{F}\left(\boldsymbol{x}^*, \boldsymbol{\lambda}\right) = \mathrm{d}_{\boldsymbol{\lambda}}\boldsymbol{y}^*\left(\boldsymbol{\lambda}\right)\partial_{\boldsymbol{y}}\bar{F}\left(\boldsymbol{x}^*, \boldsymbol{\lambda}\right) + \partial_{\boldsymbol{\lambda}}\bar{F}\left(\boldsymbol{x}^*, \boldsymbol{\lambda}\right) \approx \bar{\boldsymbol{B}}\sum_{m=0}^{M-1}\bar{\boldsymbol{A}}^m\boldsymbol{c}^{\boldsymbol{y}} + \boldsymbol{c}^{\boldsymbol{\lambda}} =: \widehat{\mathrm{d}_{\boldsymbol{\lambda}}}\bar{F}, \tag{8}$$

where, $\boldsymbol{x}^* = \begin{bmatrix}\boldsymbol{z}_1^{*\top}/\omega_1^* & \cdots & \boldsymbol{z}_n^{*\top}/\omega_n^*\end{bmatrix}^\top$, $\boldsymbol{c}^{\boldsymbol{y}} = \partial_{\boldsymbol{y}}\bar{F}\left(\boldsymbol{x}^*, \boldsymbol{\lambda}\right)$, and $\boldsymbol{c}^{\boldsymbol{\lambda}} = \partial_{\boldsymbol{\lambda}}\bar{F}\left(\boldsymbol{x}^*, \boldsymbol{\lambda}\right)$.

Estimation by recurrent backpropagation To estimate Eq. (8), we use empirical estimates $\hat{\boldsymbol{A}}^{(t)} = \boldsymbol{A}\left(\delta^{(t)}, \zeta^{(t)}\right)$ and $\hat{\boldsymbol{B}}^{(t)} = \boldsymbol{B}\left(\delta^{(t)}, \zeta^{(t)}\right)$. The following assumption ensures that $\hat{\boldsymbol{A}}^{(t)}$ and $\hat{\boldsymbol{B}}^{(t)}$ are unbiased, i.e., $\mathbb{E}_{\delta,\zeta}[\hat{\boldsymbol{A}}^{(t)}] = \bar{\boldsymbol{A}}$ and $\mathbb{E}_{\delta,\zeta}[\hat{\boldsymbol{B}}^{(t)}] = \bar{\boldsymbol{B}}$.

Assumption 4. $\delta^{(t)}$ and $\zeta^{(t)}$ are independent across the time steps $t \in \mathbb{N}$.

We estimate Eq. (8) by using the recurrent backpropagation, also referred to as the fixed-point method (Grazzi et al., 2020), to avoid the explicit computation of Jacobian matrices, $\hat{\boldsymbol{A}}^{(t)}$ and $\hat{\boldsymbol{B}}^{(t)}$. Let $\boldsymbol{u}^{(m)} = \prod_{s=0}^{m-1}\hat{\boldsymbol{A}}^{(2m+1)}\boldsymbol{c}^{\boldsymbol{y}}$ and $\boldsymbol{v}^{(m)} = \sum_{m'=0}^{m-1}\hat{\boldsymbol{B}}^{(2m')}\boldsymbol{u}^{(m')} + \boldsymbol{c}^{\boldsymbol{\lambda}}$. By initializing $\boldsymbol{u}^{(0)} \leftarrow \boldsymbol{c}^{\boldsymbol{y}}$ and $\boldsymbol{v}^{(0)} \leftarrow \boldsymbol{c}^{\boldsymbol{\lambda}}$, and by the following iterations for $m = 0, \ldots, M-1$,

$$\begin{cases}\boldsymbol{v}^{(m+1)} \leftarrow \hat{\boldsymbol{B}}^{(2m)}\boldsymbol{u}^{(m)} + \boldsymbol{v}^{(m)}, \\ \boldsymbol{u}^{(m+1)} \leftarrow \hat{\boldsymbol{A}}^{(2m+1)}\boldsymbol{u}^{(m)},\end{cases} \tag{9}$$

we obtain the hyper-gradient estimate as $\widehat{\mathrm{d}_{\boldsymbol{\lambda}}}\bar{F} \leftarrow \boldsymbol{v}^{(M)}$. In Eq. (9), we sample $\hat{\boldsymbol{B}}^{(2m)}$ and $\hat{\boldsymbol{A}}^{(2m+1)}$ in the odd- and even-rounds to ensure unbiasedness. Eq. (9) only requires Jacobian-vector products $\hat{\boldsymbol{B}}^{(2m)}\boldsymbol{u}^{(m)}$ and $\hat{\boldsymbol{A}}^{(2m+1)}\boldsymbol{u}^{(m)}$ leading $O\left(nd_{\boldsymbol{x}} + nd_{\boldsymbol{\lambda}}\right)$ and $O\left(nd_{\boldsymbol{x}}\right)$ in time, respectively.

Decentralizing backpropagation Decentralized computation of Eq. (9) requires consideration of the data locality and the communication stochasticity. From the locality of $\zeta_i^{(m)}$, clients need to communicate because only the i-th client can compute Jacobian-vector products of the i-th row block of $\hat{\boldsymbol{A}}^{(m)}$ and $\hat{\boldsymbol{B}}^{(m)}$ from their definitions Eq. (7). Moreover, we need to design a decentralized algorithm so that any required communication can be performed on stochastic and directed networks.

4.2 Hyper-Gradient Push (HGP)

We propose HGP which enables any i-th client to update their hyperparameter $\boldsymbol{\lambda}_i$ by estimating its hyper-gradient $\widehat{\mathrm{d}_{\boldsymbol{\lambda}_i}}\bar{F} = \langle \widehat{\mathrm{d}_{\boldsymbol{\lambda}}}\bar{F}\rangle_i$ over stochastic directed networks. HGP runs an unbiased alternative of Eq. (9) based on our observation that the exact computation of Eq. (9) requires undirected edges.

Exact backpropagation requires undirected edges Suppose the i-th client is responsible for computing the i-th block of $\boldsymbol{u}^{(m+1)}$ and $\boldsymbol{v}^{(m+1)}$, denoted by $\boldsymbol{u}_i^{(m+1)} \in \mathbb{R}^{d_{\boldsymbol{x}}}$ and $\boldsymbol{v}_i^{(m+1)} \in \mathbb{R}^{d_{\boldsymbol{\lambda}}}$, respectively. From Eq. (9), we obtain the following recursive iteration performed by the i-th client:

$$\begin{cases} \boldsymbol{v}_i^{(m+1)} \leftarrow \sum_{j=1}^n \delta_{i\to j}^{(2m)} \left\langle \hat{\boldsymbol{B}}^{(2m)} \right\rangle_{ij} \boldsymbol{u}_j^{(m)} + \boldsymbol{v}_i^{(m)}, \\ \boldsymbol{u}_i^{(m+1)} \leftarrow \sum_{j=1}^n \delta_{i\to j}^{(2m+1)} \left\langle \hat{\boldsymbol{A}}^{(2m+1)} \right\rangle_{ij} \boldsymbol{u}_j^{(m)}, \end{cases} \tag{10}$$

where, we use the equivalencies $\langle \hat{\boldsymbol{A}}^{(m)}\rangle_{ij} = \delta_{i\to j}^{(m)} \langle \hat{\boldsymbol{A}}^{(m)}\rangle_{ij}$ and $\langle \hat{\boldsymbol{B}}^{(m)}\rangle_{ij} = \delta_{i\to j}^{(m)} \langle \hat{\boldsymbol{B}}^{(m)}\rangle_{ij}$, as they are non-zeros only when $\delta_{i\to j}^{(m)} = 1$ from Eq. (7) and Eq. (3). To complete Eq. (10), the i-th client needs to receive $\boldsymbol{u}_j^{(m)}$ from all the j-th client with $\delta_{i\to j}^{(m)} = 1$, which is possible only when there is the communication channel from j to i (i.e., $\delta_{j\to i}^{(m)} = 1$). In other words, the exact computation of Eq. (10) is available only when the communications are undirected (i.e., $\delta_{i\to j}^{(m)} = \delta_{j\to i}^{(m)}, \forall m \in \mathbb{N}$).

Unbiased estimation via directed edges To relax the undirected communication constraint to stochastic directed communication, we propose HGP as a simple yet effective alternative of Eq. (10).

We first assumes that the i-th client knows the receiving frequency $\bar{\delta}_{j\to i} = \mathbb{E}_\delta[\delta_{j\to i}^{(m)}]$ and expected sending weight $\bar{p}_{ij} = \mathbb{E}_\delta[p_{ij}(\delta_i)]$ for all $j \in [n]$. In practice, we can estimate them through T rounds of SGP communication. We also adopt the following assumptions:

Assumption 5. If $\bar{\delta}_{j\to i} > 0$, then $\bar{\delta}_{i\to j} > 0$ and vice versa.

Assumption 6. The realization of $\delta_{i\to j}^{(m)}$ are independent over different j and i for all $m \in \mathbb{N}$.

The key idea of HGP is to replace the sending edges $\delta_{i\to j}^{(m)}$ in Eq. (10) with the debiased receiving edges $\delta_{j\to i}^{(m)}/\bar{\delta}_{j\to i}$. By initializing $\boldsymbol{u}_i^{(0)} \leftarrow \langle \boldsymbol{c}^{\boldsymbol{y}}\rangle_i = \frac{1}{n}\partial_{\boldsymbol{y}_i} F_i(\boldsymbol{x}_i^*, \boldsymbol{\lambda}_i)$ and $\boldsymbol{v}_i^{(0)} \leftarrow \langle \boldsymbol{c}^{\boldsymbol{\lambda}}\rangle_i = \frac{1}{n}\partial_{\boldsymbol{\lambda}_i} F_i(\boldsymbol{x}_i^*, \boldsymbol{\lambda}_i)$, we obtain the estimate as $\widehat{\mathrm{d}_{\boldsymbol{\lambda}_i}}\bar{F} \leftarrow \boldsymbol{v}_i^{(M)}$ after the following iterations for $m = 0, \ldots, M-1$,

$$\begin{cases} \boldsymbol{v}_i^{(m+1)} \leftarrow \sum_{j=1}^n \frac{\delta_{j\to i}^{(2m)}}{\bar{\delta}_{j\to i}} \left\langle \tilde{\boldsymbol{B}}^{(2m)} \right\rangle_{ij} \boldsymbol{u}_j^{(m)} + \boldsymbol{v}_i^{(m)}, \\ \boldsymbol{u}_i^{(m+1)} \leftarrow \sum_{j=1}^n \frac{\delta_{j\to i}^{(2m+1)}}{\bar{\delta}_{j\to i}} \left\langle \tilde{\boldsymbol{A}}^{(2m+1)} \right\rangle_{ij} \boldsymbol{u}_j^{(m)}, \end{cases} \tag{11}$$

where, $\langle \tilde{\boldsymbol{A}}^{(m)}\rangle_{ij}$ and $\langle \tilde{\boldsymbol{B}}^{(m)}\rangle_{ij}$ are defined by replacing $p_{ij}(\delta_i^{(m)})$ in Eq. (7a) and Eq. (7b) with $\bar{p}_{ij}$, respectively. The iterations above are always computable even on stochastic directed networks because the i-th client needs to receive $\boldsymbol{u}_j^{(m)}$ from the clients with $\delta_{j\to i}^{(m)} = 1$, which is always possible. We also note that Assumption 5 ensures that $\langle \tilde{\boldsymbol{A}}^{(m)}\rangle_{ij}$ and $\langle \tilde{\boldsymbol{B}}^{(m)}\rangle_{ij}$ are unbiased: $\mathbb{E}_{\delta,\zeta}[\delta_{j\to i}^{(m)}/\bar{\delta}_{j\to i}\langle \tilde{\boldsymbol{A}}^{(m)}\rangle_{ij}] = \mathbb{E}_\delta[\delta_{j\to i}^{(m)}/\bar{\delta}_{j\to i}]\mathbb{E}_{\delta,\zeta}[\langle \hat{\boldsymbol{A}}^{(m)}\rangle_{ij}] = \langle \bar{\boldsymbol{A}}\rangle_{ij}$ and the same for $\hat{\boldsymbol{B}}^{(m)}$.

HGP enjoys the same complexity as SGP in both communication and computation. HGP exchanges only $\boldsymbol{u}_i^{(\cdot)}$ having $O(d_{\boldsymbol{y}})$ in communication. In practical cases where $d_{\boldsymbol{\lambda}} \ll d_{\boldsymbol{y}}$, the Jacobian-vector products $\langle \tilde{\boldsymbol{B}}^{(\cdot)}\rangle_{ij}\boldsymbol{u}_j^{(\cdot)}$ and $\langle \tilde{\boldsymbol{A}}^{(\cdot)}\rangle_{ij}\boldsymbol{u}_j^{(\cdot)}$ are computed in $O(d_{\boldsymbol{y}})$ time.

Variance reduction We now introduce the variance-reduced version of HGP, which we call VR-HGP. The naive HGP above suffers from large variance because of $\delta_{j\to i}^{(m)}/\bar{\delta}_{j\to i}$, which can take a value far larger than one when $\bar{\delta}_{j\to i}$ is small. The multiplication of such values induces a high variance.

The idea of VR-HGP is to combine HGP with its following variant, where $\boldsymbol{w}_i^{(0)} \leftarrow \langle \boldsymbol{c}^{\boldsymbol{y}} \rangle_i$,

$$\begin{cases} \boldsymbol{v}_i^{(m+1)} = \sum_{j=1}^n \frac{\delta_{j\to i}^{(2m)}}{\bar{\delta}_{j\to i}} \left\langle \tilde{\boldsymbol{B}}^{(2m)} \right\rangle_{ij} \boldsymbol{w}_j^{(m)}, \\ \boldsymbol{w}_i^{(m+1)} = \sum_{j=1}^n \frac{\delta_{j\to i}^{(2m+1)}}{\bar{\delta}_{j\to i}} \left\langle \tilde{\boldsymbol{A}}^{(2m+1)} \right\rangle_{ij} \boldsymbol{w}_j^{(m)} + \langle \boldsymbol{c}^{\boldsymbol{y}} \rangle_i . \end{cases} \tag{12}$$

Here, $\boldsymbol{w}^{(m)}$ corresponds to the estimator of $\sum_{m'=0}^{m-1} \boldsymbol{A}^{m'} \boldsymbol{c}^{\boldsymbol{y}}$. Note that the weighted average of two different estimators results in an estimator with a smaller variance. By averaging Eq. (11) and Eq. (12) with weights $\alpha, \beta \in (0,1)$, we obtain VR-HGP as the following iterations for $m = 0, \ldots, M-1$:

$$\begin{cases} \boldsymbol{v}_i^{(m+1)} \leftarrow \alpha \left(\boldsymbol{v}_i^{(m)} + \sum_j \frac{\delta_{j\to i}^{(2m)}}{\bar{\delta}_{j\to i}} \left\langle \tilde{\boldsymbol{B}}^{(2m)} \right\rangle_{ij} \boldsymbol{u}_j^{(m)} \right) + (1-\alpha) \left(\sum_j \frac{\delta_{j\to i}^{(2m)}}{\bar{\delta}_{j\to i}} \left\langle \tilde{\boldsymbol{B}}^{(2m)} \right\rangle_{ij} \boldsymbol{w}_j^{(m)} \right), \\ \boldsymbol{u}_i^{(m+1)} \leftarrow \sum_j \frac{\delta_{j\to i}^{(2m+1)}}{\bar{\delta}_{j\to i}} \left\langle \tilde{\boldsymbol{A}}^{(2m+1)} \right\rangle_{ij} \boldsymbol{u}_j^{(m)}, \\ \boldsymbol{w}_i^{(m+1)} \leftarrow \beta \left(\sum_j \frac{\delta_{j\to i}^{(2m+1)}}{\bar{\delta}_{j\to i}} \left\langle \tilde{\boldsymbol{A}}^{(2m+1)} \right\rangle_{ij} \boldsymbol{w}_j^{(m)} + \langle \boldsymbol{c}^{\boldsymbol{y}} \rangle_i \right) + (1-\beta) \left(\boldsymbol{w}_i^{(m)} + \boldsymbol{u}_i^{(m+1)} \right), \end{cases}$$

with $\boldsymbol{v}_i^{(0)} \leftarrow \boldsymbol{0}_{d_{\boldsymbol{\lambda}}}$, $\boldsymbol{u}_i^{(0)} \leftarrow \langle \boldsymbol{c}^{\boldsymbol{y}} \rangle_i$, and $\boldsymbol{w}_i^{(0)} \leftarrow \langle \boldsymbol{c}^{\boldsymbol{y}} \rangle_i$ having the estimate as $\widehat{\mathrm{d}_{\boldsymbol{\lambda}_i}} \bar{F} \leftarrow \boldsymbol{v}_i^{(M)} + \langle \boldsymbol{c}^{\boldsymbol{\lambda}} \rangle_i$.

The following theorem provides the estimation error of the hyper-gradient using VR-HGP.

Assumption 7. $\exists \eta_A \in (0,1), \eta_B \in (0,\infty)$ such that $\forall \boldsymbol{y}_i, \boldsymbol{\lambda}_i, \zeta_i$ and $\forall i$,

$$\max\left\{ \|\partial_{\boldsymbol{y}_i} \boldsymbol{\psi}_i\|_2, \|\partial_{\boldsymbol{y}_i} \boldsymbol{\varphi}_i\|_2 \right\} \leq \frac{\eta_A}{2\kappa}, \quad \max\left\{ \|\partial_{\boldsymbol{\lambda}_i} \boldsymbol{\psi}_i\|_2, \|\partial_{\boldsymbol{\lambda}_i} \boldsymbol{\varphi}_i\|_2 \right\} \leq \frac{\eta_B}{2\kappa}, \tag{13}$$

where $\kappa = \max\left\{ n, \sum_{i,j} \frac{\bar{p}_{ji}}{\bar{\delta}_{i\to j}} \right\}$ and $\|\cdot\|_2$ denotes spectral norm.

Theorem 1 (Estimation Error of VR-HGP)**.** Suppose that Assumptions 1–7 hold true and $|\zeta_i^{(2m)}| = |\zeta_i^{(2m+1)}| = b$ for any i and m. Then, for $\alpha, \beta \in (0,1)$, with probability at least $1-\epsilon$, we have

$$\begin{aligned} &\left\| \boldsymbol{v}^{(M)} + \boldsymbol{c}^{\boldsymbol{\lambda}} - \mathrm{d}_{\boldsymbol{\lambda}} \bar{F}\left(\boldsymbol{x}^*, \boldsymbol{\lambda}\right) \right\| \\ &\leq \mu_{\alpha,\beta} \tau \sqrt{\left(\sum_{i,j=1}^n \bar{p}_{ji}^2 \left(\frac{1}{2\bar{\delta}_{i\to j}} - 1 \right)^2 + \frac{4n}{b} \right) \log \frac{n(d_{\boldsymbol{y}} + d_{\boldsymbol{\lambda}})}{\epsilon}} + \exp(-O(M)), \end{aligned}$$

where, $\|\cdot\|$ denotes ℓ_2 norm, $\exp(-O(M))$ denotes the exponentially diminishing term over M, and

$$\mu_{\alpha,\beta} = \sqrt{8 \frac{1-\alpha}{1+\alpha} \left(1 + \frac{1+\alpha(1-\beta+\beta\eta_A)}{1-\alpha(1-\beta+\beta\eta_A)} \frac{\beta^2 \eta_A^2}{1-(1-\beta+\beta\eta_A)^2} \right)}, \quad \tau = \frac{\eta_B \|\boldsymbol{c}^{\boldsymbol{y}}\|}{\kappa(1-\eta_A)}.$$

One can see that the coefficient $\mu_{\alpha,\beta}$ dominates the magnitude of the estimation error. Setting $\alpha, \beta \in (0,1)$ that minimizes $\mu_{\alpha,\beta}$ can attain a small error.[2] The proof is provided in Appendix C.

5 Related works

Personalization in federated learning We clarify the relationship between standard personalization methods and our approach by recovering them as special cases of PDBO, and by comparing the applicable communication networks.

Mansour et al. (2020) and Deng et al. (2020) propose model interpolation that provides personalized models by the optimal interpolation between the local models and the global model. This is a case

[2]Although setting $\alpha = 1$ makes $\mu_{\alpha,\beta} = 0$, the remaining error is no longer $\exp(-O(M))$ in that case. This observation implies that α slightly smaller than 1 is preferred. A similar analysis also shows that β slightly larger than 0 is preferred. Our empirical results show that $(\alpha, \beta) = (0.9, 0.1)$ performs well (appendix C.6).

of PDBO, where the inner-problem trains the global model, and the outer-problem optimizes the interpolation weight as the hyperparameter of each client. Another approach is federated multi-task learning (MTL) (Marfoq et al., 2021), which obtains personalized models by allowing clients to tune the ensemble weights of the global base-predictors. In Section 6, we empirically demonstrated that PDBO can recover the federated MTL approach by allowing the inner-problem to learn the base-predictors and the outer-problem to learn personalized ensemble weights. We see the clustering personalization (Sattler et al., 2020) as a sub-problem of federated MTL from the empirical results (Marfoq et al., 2021, J.4) that demonstrated the personalized ensemble weights recover the client clusters. Another strategy is data augmentation (Duan et al., 2019; Zhao et al., 2018), which mitigates data heterogeneity by oversampling or undersampling to train a well-generalized global model. In Section 6, we also demonstrated that PDBO can recover this approach by optimizing pseudo-sampling rates as hyperparameters. Furthermore, the generality of our framework enables us to simultaneously optimize the different types of parameters that current dedicated algorithms cannot address. In Section 6, gradient-based PDBO with a combination of different parameters yielded better performance than solely optimizing a single type of parameters.

In terms of communication networks, many personalization schemes require a centralized network (Smith et al., 2017; Sattler et al., 2020; Jiang et al., 2019), which is vulnerable to a central point of failure (Assran et al., 2019). A few fully-decentralized personalization algorithms (Marfoq et al., 2021; Lu et al., 2022) assume static and undirected networks which are not robust to failing clients and deadlocks (Tsianos et al., 2012). Althogh Vanhaesebrouck et al. (2017) and Zantedeschi et al. (2020) consider stochastic undirected settings, they can only learn simple models (linear models or a linear combination of pre-trained models). Our gradient-based PDBO can learn more complex models and run even on stochastic directed networks, which are immune to the practical problems in centralized and static undirected networks.

Distributed bilevel optimization Distributed bilevel optimizations proposed in concurrent works differ from PDBO in formulations. We categorize them into consensus distributed bilevel optimization (CDBO) (Chen et al., 2022; Tarzanagh et al., 2022; Gao et al., 2022; Yang et al., 2022) and CDBO with the local inner-problem (CDBO-Local) (Li et al., 2022; Liu et al., 2022; Lu et al., 2022).

CDBO pursues consensus also in the outer-problem, which can be obtained by imposing $\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_j$ for all $i, j \in [n]$ on PDBO outer-problem (Eq. (5-left)). Chen et al. (2022); Tarzanagh et al. (2022); Gao et al. (2022); Yang et al. (2022) applied CDBO to optimization of hyperparameter, such as L2 regularization rates. While PDBO and CDBO are inherently different tasks, both require hyper-gradient estimation over communication networks, which we discuss in the next paragraph.

CDBO-Local also requires consensus in the outer-problem, as in CDBO, whereas its inner-problem is a local optimization problem in which the optimal parameters are independent of each client. Although Lu et al. (2022) applied CDBO-Local to personalization, clients cannot benefit from others in the inner loop for better generalization. In our PDBO, both outer and inner-problems are optimized from global information; the inner parameter is trained for consensus among the clients, and the outer-parameter is optimized to improve the total performance across all clients.

We highlight that our gradient-based PDBO recovers CDBO by running SGP using the estimated hyper-gradient for the outer-problem, and recovers CDBO-Local by using SGP for the outer-problem and designing p_{ij} to form the self-loop topology in the inner SGP.

Hyper-gradient estimation over communication Networks We compare our HGP with the other hyper-gradient estimation methods performed over communication networks.

Yang et al. (2022) proposes a hyper-gradient estimation algorithm in fully-decentralized settings. However, they assume static and undirected networks, and their algorithm is complex both in computation and communication as they involve computations and communications of full Jacobians and Hessians. Tarzanagh et al. (2022) considers the hyper-gradient estimation in centralized settings, which is typical in federated learning. While their approach is advantageous in complexity because clients only compute Jacobian-vector products and exchange $O\left(d_{\boldsymbol{x}}\right)$ vectors, its applicability is tied to the centralized host-clients setting. Other CDBO methods (Chen et al., 2022; Gao et al., 2022) estimate different types of hyper-gradient. See appendix E for further details.

Our HGP enjoys reasonable complexity in computation and communication, as stated in Section 4.2, and covers a wide range of communication networks, including stochastic and directed networks.

6 EXPERIMENTS

To demonstrate the generality in personalization and applicability to practical communication environments, we introduced three different personalization approaches as special cases of gradient-based PDBO and benchmarked them with baselines on four different communication networks.

6.1 SETTINGS

We followed the settings of EMNIST classification played by $n = 100$ clients in Marfoq et al. (2021) unless otherwise mentioned. The detailed experimental settings are described in appendix D.

Communication networks We simulated four communication networks: fully-connected (`FC`), static undirected (`FixU`), stochastic undirected (`StoU`), and stochastic directed (`StoD`).

`FC` allows clients to communicate with all the others at any time step, i.e. $\delta_{i \to j}^{(t)} = 1$ for all $i, j \in [n]$ and $t \in \mathbb{N}$. `FixU` is static undirected network simulated by a binomial Erdős-Rényi graph (Erdős & Rényi, 1959) with parameter $p = 0.4$ adding the self-loop edges. Following the setting in Marfoq et al. (2021), we generated a doubly stochastic mixing matrix using the fast-mixing Markov chain (Boyd et al., 2003) rule. `StoU` simulates stochastic undirected network by letting undirected edge $\delta_{j \to i}^{(t)} = \delta_{i \to j}^{(t)}$ independently realize at each step with the probability $\bar{\delta}_{j \to i} \in [0, 1]$. In `StoD`, every direction of edges $\delta_{j \to i}^{(t)}$ is independently sampled at probability $\bar{\delta}_{j \to i}$, simulating a stochastic directed network. For all $i, j \in [n]$, $\bar{\delta}_{j \to i}$ was sampled from the uniform distribution with $[0.4, 0.8]$.

Proposed approaches We introduce and evaluated three different personalization methods as special cases of PDBO, that are, PDBO-DA, PDBO-MTL, and PDBO-MTL&DA.

PDBO-DA optimizes the pseudo-sampling rates to recover the data-augmentation-based personalization (Duan et al., 2019; Zhao et al., 2018). PDBO-DA optimizes $\boldsymbol{\lambda}_i \in \mathbb{R}^C$ to obtain the label-wise weight vector $C\texttt{Softmax}\left(\boldsymbol{\lambda}_i\right) \in [0, C]^C$. In the inner-problem, the losses of the instances labeled as $c \in [C]$ are multiplied by the c-th element of the weight vector. PDBO-MTL is obtained by formulating FedEM (Marfoq et al., 2021) as PDBO. PDBO-MTL lets each client train an ensemble classifier that outputs weighted average predictions across $K = 3$ of CNNs. PDBO-MTL trains the CNN parameters as the inner-problem and optimizes the hyperparameters $\boldsymbol{\lambda}_i \in \mathbb{R}^K$ to obtain the ensemble weight vector $\texttt{Softmax}\left(\boldsymbol{\lambda}_i\right) \in [0, 1]^K$. PDBO-MTL&DA combines PDBO-DA and PDBO-MTL by optimizing $\boldsymbol{\lambda}_i \in \mathbb{R}^{C+K}$ to obtain both the label weight and ensemble weight vectors.

Baseline approaches We compared our approaches with baselines on each communication setting. For `FC` and `FixU` settings, we compared several personalization approaches: a personalized model trained only on the local dataset (Local), FedAvg with local tuning (FedAvg+) (Jiang et al., 2019), Clustered-FL (Sattler et al., 2020), pFedMe (T Dinh et al., 2020), and centralized and decentralized versions of FedEM (Marfoq et al., 2021). We also trained the global models using SGP (Nedić & Olshevsky, 2016; Assran et al., 2019) and FedProx (Li et al., 2020). As SGP recovers FedAvg and DSGD on `FC` and `FixU`, respectively, we treat them as equivalent approaches.

Training procedure We allowed every client to generate its local dataset which has its unique label distribution, following Marfoq et al. (2021), and split it into train, validation, and test datasets.

All baselines and PDBO inner-optimizations ran the distributed learning following Marfoq et al. (2021) on `FC` and `FixU`, and ran the SGP of Eq. (4b) on `StoU` and `StoD` using the train dataset.

PDBO outer-optimizations ran 20 outer-steps tracing the average validation accuracy, and we reported the average test accuracy at an outer-step that showed the best validation accuracy. Outer-steps were performed by Adam (Kingma & Ba, 2015) from the zeros initial hyperparameters $\mathbf{0}_{d_{\boldsymbol{\lambda}}}$. To estimate the hyper-gradient for each outer-step, clients ran $M = 200$ HGP iterations with Eq. (4b) using the average cross-entropy on the train dataset as F_i. We adopted HGP for `FC` and `FixU`, and VR-HGP

Table 1: Test accuracy of personalized models on EMNIST (average clients / 10% percentile).

	Method	Communication network			
		Fully-Connected (FC)	Static Undirected (FixU)	**Stochastic Undirected (StoU)**	**Stochastic Directed (StoD)**
Global	SGP(FedAvg/DSGD)	82.2 / 73.8	82.3 / 74.1	79.7 / 71.6	79.7 / 72.5
	FedProx	69.6 / 58.2	n/a	n/a	n/a
Personalized	Local	74.7 / 63.9	74.7 / 63.9	73.7 / 63.8	73.7 / 63.8
	FedAvg+	83.0 / 75.1	n/a	n/a	n/a
	Clustered-FL	82.3 / 73.8	n/a	n/a	n/a
	pFedMe	76.2 / 65.7	n/a	n/a	n/a
	FedEM	**83.9** / 75.9	83.8 / 75.9	n/a	n/a
	PDBO-DA	82.9 / 74.8	83.0 / 75.5	80.9 / 73.2	80.8 / 72.9
	PDBO-MTL	**83.9** / **76.5**	83.9 / 76.5	81.6 / 73.8	81.6 / **75.0**
	PDBO-MTL&DA	**83.9** / 76.2	**84.0** / **77.3**	**83.0** / **76.3**	**82.2** / 74.5

with $(\alpha, \beta) = (0.9, 0.1)$ for StoU and StoD. We also made a practical modification in HGP to sample $\tilde{\boldsymbol{A}}^{(m)}$ and $\tilde{\boldsymbol{B}}^{(m)}$ together at the single m-th round, which leads to the same length of the Neumann series with the half sampling costs of the original HGP, while they are no longer unbiased.

6.2 Results and discussions

Average performance Table 1 shows the average test accuracy with weights proportional to local test dataset sizes. We observed that the ensemble-based approaches, FedEM, PDBO-MTL, and PDBO-MTL&DA performed the best on FC, and PDBO-MTL&DA outperformed on all fully-decentralized settings, that are, FixU, StoU, and StoD. Although PDBO-DA improved the average accuracy from SGP in all communication settings, it was especially effective when combined with PDBO-MTL. These results indicate that optimizing different parameters simultaneously, which is newly enabled by our PDBO, is advantageous to the personalization performance.

Fairness across the clients We also investigated whether the accuracy gain was shared among all clients. Table 1 shows the accuracy of the bottom 10% percentile of clients. All our approaches improved accuracy at the 10% percentile from global model approaches (SGP and FedProx) in all communication settings, confirming that the clients fairly benefited from our personalization.

Applicability to stochastic communication networks The communication network limits the available personalization methods, especially when the network is stochastic. Although FedEM is one of the few personalization methods feasible in fully-decentralized settings, it requires the doubly stochastic mixing matrix to be known, which is impractical on stochastic networks (Tsianos et al., 2012). As PDBO encompasses SGP as the inner-problem and HGP can run on stochastic communication networks, our approaches succeeded in personalizing on StoU and StoD.

Robustness to communication directionality Our HGP and VR-HGP estimate hyper-gradient solely from the directed communication edges, rather than running the standard recurrent backpropagation which requires undirected edges. The improvement in our approaches on StoD demonstrated that VR-HGP estimated the hyper-gradient with sufficiently small errors to solve PDBO.

7 Conclusion

This study proposed a gradient-based PDBO, which reduces most personalization approaches to the optimization of hyperparameters possessed by each client. We also proposed HGP that estimates the hyper-gradient through communications over stochastic and directed communication networks. In addition, we introduced a variance-reduced HGP that mitigated the estimation variance caused by the stochasticity of communication edges and provided its theoretical error bound. Our empirical results demonstrated that our gradient-based PDBO with HGP enabled combining different personalization approaches which led to state-of-the-art performance, and it performed on different simulated communication environments including a stochastic and directed network.

REPRODUCIBILITY STATEMENT

We provide the detailed experiment settings of Section 6 in appendix D including any modification to the benchmark conducted by Marfoq et al. (2021) and will be releasing their implementations after the review process. Our theoretical contributions and required assumptions are stated in Section 4.2. The detailed derivations of our HGP and VR-HGP is provided in appendix A and appendix B, respectively. We also provide detailed proof of the estimation error bound of VR-HGP in appendix C.

REFERENCES

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Mike Rabbat. Stochastic gradient push for distributed deep learning. In *International Conference on Machine Learning*, 2019.

Pascal Bianchi, Gersende Fort, and Walid Hachem. Performance of a distributed stochastic approximation algorithm. *IEEE Transactions on Information Theory*, 59(11):7405–7418, 2013.

Stephen Boyd, Persi Diaconis, and Lin Xiao. Fastest mixing markov chain on a graph. *SIAM REVIEW*, 46:667–689, 2003.

Duc Bui, Kshitiz Malik, Jack Goetz, Honglei Liu, Seungwhan Moon, Anuj Kumar, and Kang G Shin. Federated user representation learning. *arXiv preprint arXiv:1909.12535*, 2019.

Xuxing Chen, Minhui Huang, and Shiqian Ma. Decentralized bilevel optimization. *arXiv preprint arXiv:2206.05670*, 2022.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. EMNIST: Extending MNIST to handwritten letters. In *International Joint Conference on Neural Networks*, 2017.

Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

Moming Duan, Duo Liu, Xianzhang Chen, Yujuan Tan, Jinting Ren, Lei Qiao, and Liang Liang. Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications. In *IEEE 37th International Conference on Computer Design*, 2019.

Paul Erdős and Alfréd Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

Hongchang Gao, Bin Gu, and My T Thai. Stochastic bilevel distributed optimization over a network. *arXiv preprint arXiv:2206.15025*, 2022.

Riccardo Grazzi, Luca Franceschi, Massimiliano Pontil, and Saverio Salzo. On the iteration complexity of hypergradient computation. In *International Conference on Machine Learning*, 2020.

Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Junyi Li, Feihu Huang, and Heng Huang. Local stochastic bilevel optimization with momentum-based variance reduction. *arXiv preprint arXiv:2205.01608*, 2022.

Mu Li, David G. Andersen, Jun Woo Park, Alexander J. Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J. Shekita, and Bor-Yiing Su. Scaling distributed machine learning with the parameter server. In *11th USENIX Symposium on Operating Systems Design and Implementation*, 2014.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In I. Dhillon, D. Papailiopoulos, and V. Sze (eds.), *Proceedings of Machine Learning and Systems*, 2020.

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, 2017.

Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. Asynchronous decentralized parallel stochastic gradient descent. In *International Conference on Machine Learning*, 2018.

Zhuqing Liu, Xin Zhang, Prashant Khanduri, Songtao Lu, and Jia Liu. Interact: Achieving low sample and communication complexities in decentralized bilevel learning over networks. *arXiv preprint arXiv:2207.13283*, 2022.

Songtao Lu, Xiaodong Cui, Mark S Squillante, Brian Kingsbury, and Lior Horesh. Decentralized bilevel optimization for personalized client learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022.

Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Federated multi-task learning under a mixture of distributions. In *Advances in Neural Information Processing Systems*, 2021.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, 2017.

Angelia Nedić and Alex Olshevsky. Stochastic gradient-push for strongly convex functions on time-varying directed graphs. *IEEE Transactions on Automatic Control*, 61(12):3936–3947, 2016.

Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, 2017.

Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. In *Advances in Neural Information Processing Systems*, 2020.

Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Davoud Ataee Tarzanagh, Mingchen Li, Christos Thrampoulidis, and Samet Oymak. Fednest: Federated bilevel, minimax, and compositional optimization. *arXiv preprint arXiv:2205.02215*, 2022.

Konstantinos I Tsianos, Sean Lawlor, and Michael G Rabbat. Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning. In *50th Annual Allerton Conference on Communication, Control, and Computing*, 2012.

Paul Vanhaesebrouck, Aurélien Bellet, and Marc Tommasi. Decentralized collaborative learning of personalized models over networks. In *Artificial Intelligence and Statistics*, 2017.

Shuoguang Yang, Xuezhou Zhang, and Mengdi Wang. Decentralized gossip-based stochastic bilevel optimization over communication networks. *arXiv preprint arXiv:2206.10870*, 2022.

Valentina Zantedeschi, Aurélien Bellet, and Marc Tommasi. Fully decentralized joint learning of personalized models and collaboration graphs. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

A ESTIMATION OF HYPER-GRADIENT

Notation For a vector $v \in \mathbb{R}^d$, $\|v\| = \sqrt{\sum_{i=1}^d v_i^2}$ is its ℓ_2 norm. For a matrix $V \in \mathbb{R}^{d_1 \times d_2}$, $\|V\|_2$ is its largest singular value.

A.1 STATIONARITY OF SGP

We consider the generalized version of SGP over n nodes as follows:

$$\boldsymbol{y}_i^{(t+1)} = \sum_{j=1}^n p_{ji}(\delta_j^{(t)}) \boldsymbol{\varphi}_j \left(\boldsymbol{y}_j^{(t)}; \boldsymbol{\lambda}_j, \zeta_j^{(t)}\right) + \boldsymbol{\psi}_i \left(\boldsymbol{y}_i^{(t)}; \boldsymbol{\lambda}_i, \zeta_i^{(t)}\right), \;\; p_{ji}(\delta_j^{(t)}) = \delta_{j\to i}^{(t)} p_{ji}\left(\delta_{j\to 1}^{(t)}, \ldots, \delta_{j\to n}^{(t)}\right),$$

We set $\delta_{i\to i}^{(t)} = 1$ for all i and t, i.e., every client can send a message to itself at any time step.

Assumption 2 ensures $\boldsymbol{y}^* = \boldsymbol{y}^{(T)}$ is the stationary point:

$$\begin{aligned}\boldsymbol{y}_i^* &= \mathbb{E}_{\delta,\zeta}\left[\sum_{j=1}^n p_{ji}\boldsymbol{\varphi}_j\left(\boldsymbol{y}_j^*; \boldsymbol{\lambda}_j, \zeta_j\right) + \boldsymbol{\psi}_i\left(\boldsymbol{y}_i^*; \boldsymbol{\lambda}_i, \zeta_i\right)\right] \\ &= \sum_{j=1}^n \bar{p}_{ji}\mathbb{E}_\zeta\left[\boldsymbol{\varphi}_j\left(\boldsymbol{y}_j^*; \boldsymbol{\lambda}_j, \zeta_j\right)\right] + \mathbb{E}_\zeta\left[\boldsymbol{\psi}_i\left(\boldsymbol{y}_i^*; \boldsymbol{\lambda}_i, \zeta_i\right)\right],\end{aligned}$$

where $\bar{p}_{ji} = \mathbb{E}_\delta\left[p_{ji}\right]$.

A.2 HYPER-GRADIENT BY IMPLICIT DIFFERENTIATION

We adopt Assumption 1 so that $\boldsymbol{\varphi}, \boldsymbol{\psi}$, and $\bar{F}$ to be differentiable. The differentiation of $\boldsymbol{y}_i^*$ by $\boldsymbol{\lambda}_j$ is

$$\begin{aligned}\mathrm{d}_{\boldsymbol{\lambda}_j}\boldsymbol{y}_i^* = \sum_{j=1}^n \bar{p}_{ji}&\left(\mathrm{d}_{\boldsymbol{\lambda}_j}\boldsymbol{y}_i^*\partial_{\boldsymbol{y}_i}\mathbb{E}_\zeta\left[\boldsymbol{\varphi}_j\left(\boldsymbol{y}_j^*; \boldsymbol{\lambda}_j, \zeta_j\right)\right] + \partial_{\boldsymbol{\lambda}_j}\mathbb{E}_\zeta\left[\boldsymbol{\varphi}_j\left(\boldsymbol{y}_j^*; \boldsymbol{\lambda}_j, \zeta_j\right)\right]\right) \\ &+ \mathbb{1}_{ji}\left(\mathrm{d}_{\boldsymbol{\lambda}_j}\boldsymbol{y}_i^*\partial_{\boldsymbol{y}_i}\mathbb{E}_\zeta\left[\boldsymbol{\psi}_i\left(\boldsymbol{y}_i^*; \boldsymbol{\lambda}_i, \zeta_i\right)\right] + \partial_{\boldsymbol{\lambda}_i}\mathbb{E}_\zeta\left[\boldsymbol{\psi}_i\left(\boldsymbol{y}_i^*; \boldsymbol{\lambda}_i, \zeta_i\right)\right]\right).\end{aligned}$$

Let $\boldsymbol{y}^* = [\boldsymbol{y}_i^*]_i$ and $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_i]_i$ be the concatenated parameters and hyperparameters, respectively. We can write the differentiation in the matrix form by

$$\mathrm{d}_{\boldsymbol{\lambda}}\boldsymbol{y}^* = \mathrm{d}_{\boldsymbol{\lambda}}\boldsymbol{y}^*\boldsymbol{A} + \boldsymbol{B},$$

where

$$\begin{aligned}\boldsymbol{A} &= \left[\mathbb{1}_{ji}\boldsymbol{A}_i^\psi + \bar{p}_{ji}\boldsymbol{A}_j^\varphi\right]_{ji} \in \mathbb{R}^{nd_{\boldsymbol{y}} \times nd_{\boldsymbol{y}}}, \\ &\boldsymbol{A}_i^\psi = \partial_{\boldsymbol{y}_i}\mathbb{E}_\zeta\left[\boldsymbol{\varphi}_i(\boldsymbol{y}_i^*; \boldsymbol{\lambda}_i, \zeta_i)\right] \in \mathbb{R}^{d_{\boldsymbol{y}} \times d_{\boldsymbol{y}}}, \qquad \boldsymbol{A}_j^\varphi = \partial_{\boldsymbol{y}_j}\mathbb{E}_\zeta\left[\boldsymbol{\varphi}_j(\boldsymbol{y}_j^*; \boldsymbol{\lambda}_j, \zeta_j)\right] \in \mathbb{R}^{d_{\boldsymbol{y}} \times d_{\boldsymbol{y}}}, \\ \boldsymbol{B} &= \left[\mathbb{1}_{ji}\boldsymbol{B}_i^\psi + \bar{p}_{ji}\boldsymbol{B}_j^\varphi\right]_{ji} \in \mathbb{R}^{nd_{\boldsymbol{\lambda}} \times nd_{\boldsymbol{y}}}, \\ &\boldsymbol{B}_i^\psi = \partial_{\boldsymbol{\lambda}_i}\mathbb{E}_\zeta\left[\boldsymbol{\varphi}_i(\boldsymbol{y}_i^*; \boldsymbol{\lambda}_i, \zeta_i)\right] \in \mathbb{R}^{d_{\boldsymbol{\lambda}} \times d_{\boldsymbol{y}}}, \qquad \boldsymbol{B}_j^\varphi = \partial_{\boldsymbol{\lambda}_j}\mathbb{E}_\zeta\left[\boldsymbol{\varphi}_j(\boldsymbol{y}_j^*; \boldsymbol{\lambda}_j, \zeta_j)\right] \in \mathbb{R}^{d_{\boldsymbol{\lambda}} \times d_{\boldsymbol{y}}}.\end{aligned}$$

Then, we have

$$\mathrm{d}_{\boldsymbol{\lambda}}\boldsymbol{y}^* = \boldsymbol{B}(\boldsymbol{I} - \boldsymbol{A})^{-1}.$$

In particular, we have

$$\mathrm{d}_{\boldsymbol{\lambda}_j}\boldsymbol{y}_i^* = \sum_k \langle \boldsymbol{B}\rangle_{jk}\langle(\boldsymbol{I} - \boldsymbol{A})^{-1}\rangle_{ki},$$

where $\langle\cdot\rangle_{jk}$ and $\langle\cdot\rangle_{ki}$ denotes the (j,k)-th and (k,i)-th block of the matrix.

The hyper-gradient of the objective function $F(\boldsymbol{x}^*, \boldsymbol{\lambda}) = \sum_i F_i(\boldsymbol{x}_i^*, \boldsymbol{\lambda}_i)$ is then given as

$$\begin{aligned}\mathrm{d}_{\boldsymbol{\lambda}_j}F(\boldsymbol{y}^*, \boldsymbol{\lambda}) &= \sum_i \mathrm{d}_{\boldsymbol{\lambda}_j}\boldsymbol{y}_i^* \underbrace{\partial_{\boldsymbol{y}_i}F_i(\boldsymbol{y}_i^*, \boldsymbol{\lambda}_i)}_{\boldsymbol{c}_i^{\boldsymbol{y}}} + \underbrace{\partial_{\boldsymbol{\lambda}_j}F_j(\boldsymbol{y}_j^*, \boldsymbol{\lambda}_j)}_{\boldsymbol{c}_j^{\boldsymbol{\lambda}}} \\ &= \sum_i \mathrm{d}_{\boldsymbol{\lambda}_j}\boldsymbol{y}_i^*\boldsymbol{c}_i^{\boldsymbol{y}} + \boldsymbol{c}_j^{\boldsymbol{\lambda}}.\end{aligned}$$

A.3 ESTIMATION OF HYPERGRADIENT

In the remainder, we consider $\boldsymbol{\psi}_i$ and $\boldsymbol{\varphi}_j$ of the following forms:

$$\boldsymbol{\psi}_i(\boldsymbol{y}_i;\boldsymbol{\lambda}_i,\zeta_i) = \frac{1}{|\zeta_i|}\sum_{\xi\in\zeta_i} g_i(\boldsymbol{y}_i;\boldsymbol{\lambda}_i,\xi)$$
$$\boldsymbol{\varphi}_j(\boldsymbol{y}_j;\boldsymbol{\lambda}_j,\zeta_j) = \frac{1}{|\zeta_j|}\sum_{\xi\in\zeta_i} h_j(\boldsymbol{y}_j;\boldsymbol{\lambda}_j,\xi),$$

for some $g_i(\cdot;\boldsymbol{\lambda}_i,\xi):\mathbb{R}^{d_{\boldsymbol{y}}}\to\mathbb{R}^{d_{\boldsymbol{y}}}$ and $h_j(\cdot;\boldsymbol{\lambda}_j,\xi):\mathbb{R}^{d_{\boldsymbol{y}}}\to\mathbb{R}^{d_{\boldsymbol{y}}}$, which are true for SGP in Eq. (4a) and Eq. (4b). Assumption 1 ensures that g_i and h_j are differentiable with respect to both $\boldsymbol{y}$ and $\boldsymbol{\lambda}$.

A.3.1 ESTIMATION OF A AND B

Because the matrices A and B are defined as the expectation over the data minibatch ζ_i,ζ_j as well as the realization of communication network δ, we estimate them from the observation as follows.

$$\hat{\boldsymbol{A}} = \left[\mathbb{1}_{ji}\hat{\boldsymbol{A}}_i^{\psi} + \bar{p}_{ji}\hat{\boldsymbol{A}}_j^{\varphi}\right]_{ji} \in \mathbb{R}^{nd_{\boldsymbol{y}}\times nd_{\boldsymbol{y}}},$$
$$\hat{\boldsymbol{A}}_i^{\psi} = \frac{1}{|\zeta_i|}\sum_{\xi\in\zeta_i}\partial_{\boldsymbol{y}_i} g_i(\boldsymbol{y}_i^*;\boldsymbol{\lambda}_i,\xi)\in\mathbb{R}^{d_{\boldsymbol{y}}\times d_{\boldsymbol{y}}},\quad \hat{\boldsymbol{A}}_j^{\varphi} = \frac{1}{|\zeta_j|}\sum_{\xi\in\zeta_j}\partial_{\boldsymbol{y}_j} h_j(\boldsymbol{y}_j^*;\boldsymbol{\lambda}_j,\xi)\in\mathbb{R}^{d_{\boldsymbol{y}}\times d_{\boldsymbol{y}}},$$
$$\hat{\boldsymbol{B}} = \left[\mathbb{1}_{ji}\hat{\boldsymbol{B}}_i^{\psi} + \bar{p}_{ji}\hat{\boldsymbol{B}}_j^{\varphi}\right]_{ji} \in \mathbb{R}^{nd_{\boldsymbol{\lambda}}\times nd_{\boldsymbol{y}}},$$
$$\hat{\boldsymbol{B}}_i^{\psi} = \frac{1}{|\zeta_i|}\sum_{\xi\in\zeta_i}\partial_{\boldsymbol{\lambda}_i} h_i(\boldsymbol{y}_i^*;\boldsymbol{\lambda}_i,\xi)\in\mathbb{R}^{d_{\boldsymbol{\lambda}}\times d_{\boldsymbol{y}}},\quad \hat{\boldsymbol{B}}_j^{\varphi} = \frac{1}{|\zeta_j|}\sum_{\xi\in\zeta_j}\partial_{\boldsymbol{\lambda}_j} \boldsymbol{\varphi}_{ji}(\boldsymbol{y}_j^*;\boldsymbol{\lambda}_j,\xi)\in\mathbb{R}^{d_{\boldsymbol{\lambda}}\times d_{\boldsymbol{y}}}.$$

A.3.2 APPROXIMATION BY NEUMANN SERIES

With Assumption 3, we have $\|\boldsymbol{A}\|_2 < 1$ We can thus approximate $(\boldsymbol{I}-\boldsymbol{A})^{-1}$ by the truncated Neumann series up to the M-th term as

$$(\boldsymbol{I}-\boldsymbol{A})^{-1} = \sum_{m=0}^{\infty}\boldsymbol{A}^m \approx \sum_{m=0}^{M-1}\boldsymbol{A}^m.$$

The approximation of the hyper-gradient could be expressed as

$$\mathrm{d}_{\boldsymbol{\lambda}}F(\boldsymbol{x}^*,\boldsymbol{\lambda}) \approx \boldsymbol{B}\sum_{m=0}^{M-1}\boldsymbol{A}^m\boldsymbol{c}_i^{\boldsymbol{y}} + \boldsymbol{c}_j^{\boldsymbol{\lambda}}.$$

By replacing A and B with the estimators $\hat{A}$ and $\hat{B}$, we have

$$\mathrm{d}_{\boldsymbol{\lambda}}\bar{F}\left(\boldsymbol{x}^*,\boldsymbol{\lambda}\right) \approx \sum_{m=0}^{M-1}\hat{\boldsymbol{B}}^{(2m)}\prod_{s=0}^{m-1}\hat{\boldsymbol{A}}^{(2s+1)}\boldsymbol{c}^{\boldsymbol{y}} + \boldsymbol{c}^{\boldsymbol{\lambda}},$$

where $\hat{A}^{(2s+1)}$ and $\hat{B}^{(2m)}$ denotes the estimators at the $2s+1$-th and the $2m$-th step of the communication round, respectively. In this estimator, we estimate A in the odd-numbered steps and estimate B in the even-numbered steps of the communication round, respectively.

A.4 HYPER-GRADIENT PUSH (HGP)

We now present our proposed method, hyper-gradient push (HGP), which is a modified version of the recurrent backpropagation. HGP can run even on stochastic and directed networks while enjoying the same order of communication efficiency as SGP. In HGP, we adopt Assumptions 5 and 6, and assume that $\{\bar{\delta}_{j\to i} = \mathbb{E}_{\delta}\left[\delta_{j\to i}\right]\}_{j,i}$ and $\{\bar{p}_{ji}\}_{j,i}$ are known.

The idea of HGP is to use $\delta_{ij}\frac{\bar{p}_{ji}}{\bar{\delta}_{ij}}$ instead of p_{ji} in $\hat{A}$ and $\hat{B}$ as follows.

$$\hat{\boldsymbol{A}} = \left[\mathbb{1}_{ji}\hat{\boldsymbol{A}}_i^{\psi} + \delta_{i\to j}\frac{\bar{p}_{ji}}{\bar{\delta}_{i\to j}}\hat{\boldsymbol{A}}_j^{\varphi}\right]_{ji},$$
$$\hat{\boldsymbol{B}} = \left[\mathbb{1}_{ji}\hat{\boldsymbol{B}}_i^{\psi} + \delta_{i\to j}\frac{\bar{p}_{ji}}{\bar{\delta}_{i\to j}}\hat{\boldsymbol{B}}_j^{\varphi}\right]_{ji}.$$

Under Assumption 4 where δ and ζ are independent, these are the unbiased estimators because

$$\mathbb{E}_{\delta,\zeta}\left[\hat{\boldsymbol{A}}\right] = \left[\mathbb{1}_{ji}\underbrace{\mathbb{E}_{\zeta_i}\left[\hat{\boldsymbol{A}}_i^{\psi}\right]}_{=\boldsymbol{A}_i^{\psi}} + \underbrace{\mathbb{E}_{\delta_{i\to j}}\left[\delta_{i\to j}\frac{\bar{p}_{ji}}{\bar{\delta}_{i\to j}}\right]}_{\bar{\delta}_{i\to j}\frac{\bar{p}_{ji}}{\bar{\delta}_{i\to j}}=\bar{p}_{ji}}\underbrace{\mathbb{E}_{\zeta_j}\left[\hat{\boldsymbol{A}}_j^{\varphi}\right]}_{=\boldsymbol{A}_j^{\varphi}}\right]_{ji}$$
$$= \left[\mathbb{1}_{ji}\boldsymbol{A}_i^{\psi} + \bar{p}_{ji}\boldsymbol{A}_j^{\varphi}\right]_{ji} = \boldsymbol{A},$$
$$\mathbb{E}_{\delta,\zeta}\left[\hat{\boldsymbol{B}}\right] = \left[\mathbb{1}_{ji}\underbrace{\mathbb{E}_{\zeta_i}\left[\hat{\boldsymbol{B}}_i^{\psi}\right]}_{=\boldsymbol{B}_i} + \underbrace{\mathbb{E}_{\delta_{i\to j}}\left[\delta_{i\to j}\frac{\bar{p}_{ji}}{\bar{\delta}_{i\to j}}\right]}_{\bar{\delta}_{i\to j}\frac{\bar{p}_{ji}}{\bar{\delta}_{i\to j}}=\bar{p}_{ji}}\underbrace{\mathbb{E}_{\zeta_j}\left[\hat{\boldsymbol{B}}_j^{\varphi}\right]}_{=\boldsymbol{B}_j^{\varphi}}\right]_{ji}$$
$$= \left[\mathbb{1}_{ji}\boldsymbol{B}_i^{\psi} + \bar{p}_{ji}\boldsymbol{B}_j^{\varphi}\right]_{ji} = \boldsymbol{B}.$$

Recall that the hyper-gradient can be approximated as

$$\mathrm{d}_{\boldsymbol{\lambda}_j}F(\boldsymbol{x}^*,\boldsymbol{\lambda}) \approx \sum_{m=0}^{M-1}\sum_k \langle \boldsymbol{B}\rangle_{jk}\sum_i \langle \boldsymbol{A}^m\rangle_{ki}\boldsymbol{c}_i^{\boldsymbol{y}} + \boldsymbol{c}_j^{\boldsymbol{\lambda}}. \tag{14}$$

By replacing the expectation with the above estimators $\hat{A}$ and $\hat{B}$, we have

$$\widehat{\mathrm{d}_{\boldsymbol{\lambda}_j}}F(\boldsymbol{x}^*,\boldsymbol{\lambda}) = \sum_{m=0}^{M-1}\sum_k \left\langle \hat{\boldsymbol{B}}^{(2m)}\right\rangle_{jk}\sum_i \left\langle \prod_{s=0}^{m-1}\hat{\boldsymbol{A}}^{(2s+1)}\right\rangle_{ki}\boldsymbol{c}_i^{\boldsymbol{y}} + \boldsymbol{c}_j^{\boldsymbol{\lambda}}. \tag{15}$$

Let $\boldsymbol{u}_k^{(m)} = \sum_i \left\langle \prod_{s=0}^{m-1}\hat{\boldsymbol{A}}^{(2s+1)}\right\rangle_{ki}\boldsymbol{c}_i^{\boldsymbol{y}}$. We note that $\boldsymbol{u}_k^{(m+1)}$ can be computed recursively as

$$\boldsymbol{u}_k^{(m+1)} = \sum_{k'}\left\langle \hat{\boldsymbol{A}}^{(2m+1)}\right\rangle_{kk'}\boldsymbol{u}_{k'}^{(m)}.$$

By using this fact, we can rewrite the estimator as

$$\widehat{\mathrm{d}_{\boldsymbol{\lambda}_j}}F(\boldsymbol{x}^*,\boldsymbol{\lambda}) = \sum_{m=0}^{M-1}\sum_k \left\langle \hat{\boldsymbol{B}}^{(2m)}\right\rangle_{jk}\sum_{k'}\left\langle \hat{\boldsymbol{A}}^{(2m-1)}\right\rangle_{kk'}\underbrace{\sum_i\left\langle \prod_{s=0}^{m-2}\hat{\boldsymbol{A}}^{(2s+1)}\right\rangle_{k'i}\boldsymbol{c}_i^{\boldsymbol{y}}}_{\boldsymbol{u}_{k'}^{(m-1)}} + \boldsymbol{c}_j^{\boldsymbol{\lambda}}$$
$$= \sum_{m=0}^{M-1}\sum_k \left\langle \hat{\boldsymbol{B}}^{(2m)}\right\rangle_{jk}\underbrace{\sum_{k'}\left\langle \hat{\boldsymbol{A}}^{(2m-1)}\right\rangle_{kk'}\boldsymbol{u}_{k'}^{(m-1)}}_{\boldsymbol{u}_k^{(m)}} + \boldsymbol{c}_j^{\boldsymbol{\lambda}}.$$

We can then derive the proposed algorithm, hyper-gradient push, as follows:

Hyper-Gradient Push (HGP)

$\boldsymbol{u}_j^{(0)} \leftarrow \boldsymbol{c}_j^{\boldsymbol{y}}$, $\boldsymbol{v}_j^{(0)} \leftarrow \boldsymbol{c}_j^{\boldsymbol{\lambda}}$

$$\begin{cases} \boldsymbol{v}_j^{(m+1)} \leftarrow \boldsymbol{v}_j^{(m)} + \sum_k \left\langle \hat{\boldsymbol{B}}^{(2m)} \right\rangle_{jk} \boldsymbol{u}_k^{(m)} \\ \boldsymbol{u}_k^{(m+1)} \leftarrow \sum_{k'} \left\langle \hat{\boldsymbol{A}}^{(2m+1)} \right\rangle_{kk'} \boldsymbol{u}_{k'}^{(m)} \end{cases}$$

for $m = 0, 1, 2, \ldots, M-1$

$\widehat{\mathrm{d}_{\boldsymbol{\lambda}_i}} F(\boldsymbol{x}^*, \boldsymbol{\lambda}) \leftarrow \boldsymbol{v}_j^{(M)}$

In HGP, the estimator could be obtained after the $2M$ rounds of communication. In each round of the communication, the clients communicate $\boldsymbol{u}_k^{(m)} \in \mathbb{R}^{d_{\boldsymbol{y}}}$ which is $O(d_{\boldsymbol{y}})$ parameters only, the same as the standard communication for SGP update.

B Variance Reduction

We now introduce the variance-reduced version of HGP. The naive HGP above suffers from the large variance because of $\delta_{j\to i}^{(m)}/\bar{\delta}_{j\to i}$; this term can take a value far larger than one when $\bar{\delta}_{j\to i}$ is small. The multiplication of such values induces high variance.

Recall that, in HGP, we aim at approximating the estimator

$$\mathrm{d}_{\boldsymbol{\lambda}_j} F(\boldsymbol{x}^*, \boldsymbol{\lambda}) \approx \sum_{m=0}^{M-1} \sum_k \langle \boldsymbol{B} \rangle_{jk} \sum_i \langle \boldsymbol{A}^m \rangle_{ki} \boldsymbol{c}_i^{\boldsymbol{y}} + \boldsymbol{c}_j^{\boldsymbol{\lambda}}.$$

With $\boldsymbol{v}_j^{(0)} \leftarrow 0$, $\boldsymbol{u}_k^{(0)} \leftarrow C_k^{\boldsymbol{y}}$, HGP computes the first term of the right-hand-side by iterating

$$\begin{aligned} \boldsymbol{v}_j^{(m+1)} &\leftarrow \boldsymbol{v}_j^{(m)} + \sum_k \langle \boldsymbol{B} \rangle_{jk} \boldsymbol{u}_k^{(m)}, \\ \boldsymbol{u}_k^{(m+1)} &\leftarrow \sum_{k'} \langle \boldsymbol{A} \rangle_{kk'} \boldsymbol{u}_{k'}^{(m)}, \end{aligned}$$

where $u_k^{(m+1)}$ is equivalent to $\sum_{k'} \langle A^{m+1} \rangle_{kk'} \boldsymbol{c}_{k'}^{\boldsymbol{y}}$. We can also consider another way of computing the first term. With $\boldsymbol{v}_j^{(0)} \leftarrow \boldsymbol{0}_{d_{\boldsymbol{\lambda}}}$, $\boldsymbol{w}_k^{(0)} \leftarrow \boldsymbol{c}_k^{\boldsymbol{y}}$, we can compute

$$\begin{aligned} \boldsymbol{v}_j^{(m+1)} &\leftarrow \sum_k \langle \boldsymbol{B} \rangle_{jk} \boldsymbol{w}_k^{(m)}, \\ \boldsymbol{w}_k^{(m+1)} &\leftarrow \sum_{k'} \langle \boldsymbol{A} \rangle_{kk'} \boldsymbol{w}_{k'}^{(m)} + \boldsymbol{c}_k^{\boldsymbol{y}}, \end{aligned}$$

where $\boldsymbol{w}_k^{(m+1)}$ is equivalent to $\sum_{m'=0}^{m+1} \sum_{k'} \left\langle \boldsymbol{A}^{m'} \right\rangle_{kk'} \boldsymbol{c}_{k'}^{\boldsymbol{y}} = \sum_{m'=0}^{m+1} \boldsymbol{u}_k^{(m')} = \boldsymbol{w}_k^{(m)} + \boldsymbol{u}_k^{(m+1)}$.

By combining the above two formulas, we can derive the general expression of HGP as

$$\begin{aligned} \boldsymbol{v}_j^{(m+1)} &\leftarrow \alpha \left(\boldsymbol{v}_j^{(m)} + \sum_k \langle \boldsymbol{B} \rangle_{jk} \boldsymbol{u}_k^{(m)} \right) + (1-\alpha) \sum_k \langle \boldsymbol{B} \rangle_{jk} \boldsymbol{w}_k^{(m)}, \\ \boldsymbol{u}_k^{(m+1)} &\leftarrow \sum_{k'} \langle \boldsymbol{A} \rangle_{kk'} \boldsymbol{u}_{k'}^{(m)}, \\ \boldsymbol{w}_k^{(m+1)} &\leftarrow \beta \left(\sum_{k'} \langle \boldsymbol{A} \rangle_{kk'} \boldsymbol{w}_{k'}^{(m)} + \boldsymbol{c}_k^{\boldsymbol{y}} \right) + (1-\beta) \left(\boldsymbol{w}_k^{(m)} + \boldsymbol{u}_k^{(m+1)} \right), \end{aligned}$$

where $\alpha, \beta \in [0, 1]$ are the interpolation weights. By replacing $\boldsymbol{A}$, $\boldsymbol{B}$ by the empirical estimates $\hat{\boldsymbol{A}}$, $\hat{\boldsymbol{B}}$, we obtain the general expression of HGP as follows.

General HGP for Variance Reduction

$$\boldsymbol{v}_j^{(0)} \leftarrow \mathbf{0}_{d_{\boldsymbol{\lambda}}},\ \boldsymbol{u}_j^{(0)} \leftarrow \boldsymbol{c}_j^{\boldsymbol{y}},\ \boldsymbol{w}_j^{(0)} \leftarrow \boldsymbol{c}_j^{\boldsymbol{y}}$$

$$\begin{cases} \boldsymbol{v}_j^{(m+1)} \leftarrow \alpha\left(\boldsymbol{v}_j^{(m)} + \sum_k \left\langle \hat{\boldsymbol{B}}^{(2m)} \right\rangle_{jk} \boldsymbol{u}_k^{(m)}\right) + (1-\alpha)\sum_k \left\langle \hat{\boldsymbol{B}}^{(2m)} \right\rangle_{jk} \boldsymbol{w}_k^{(m)} \\ \boldsymbol{u}_k^{(m+1)} \leftarrow \sum_{k'} \left\langle \hat{\boldsymbol{A}}^{(2m+1)} \right\rangle_{kk'} \boldsymbol{u}_{k'}^{(m)} \\ \boldsymbol{w}_k^{(m+1)} \leftarrow \beta\left(\sum_{k'} \left\langle \hat{\boldsymbol{A}}^{(2m+1)} \right\rangle_{kk'} \boldsymbol{w}_{k'}^{(m)} + \boldsymbol{c}_k^{\boldsymbol{y}}\right) + (1-\beta)\left(\boldsymbol{w}_k^{(m)} + \boldsymbol{u}_k^{(m+1)}\right), \end{cases}$$

for $m = 0, 1, 2, \ldots, M-1$

$$\widehat{\mathrm{d}_{\boldsymbol{\lambda}_i}} F(\boldsymbol{x}^*, \boldsymbol{\lambda}) \leftarrow \boldsymbol{v}_j^{(M)} + \boldsymbol{c}_j^{\boldsymbol{\lambda}}$$

We note that this general HGP is the weighted average of the two different estimation algorithms, which results in an estimator with a smaller variance. That is, by choosing $\alpha, \beta \in [0, 1]$ appropriately, we can obtain an estimate of the hyper-gradient with a smaller variance. From the computational perspective, this general HGP has properties similar to the original HGP: it can be computed even on stochastic and directed networks; the estimator could be obtained after the $2M$ rounds of communication; and the clients communicate $O(d_{\boldsymbol{y}})$ parameters in each iteration.

C Estimation Error of Hyper-Gradient

In the following, we assume that the derivatives of g_i and h_j are bounded.

Assumption 8. $\exists \eta_A \in (0,1), \eta_B \in (0,\infty)$ such that $\forall \xi$ and $\forall i, j$,

$$\max\left\{\sup_{\boldsymbol{y}_i,\boldsymbol{\lambda}_i,\xi} \|\partial_{\boldsymbol{y}_i} g_i(\boldsymbol{y}_i,\boldsymbol{\lambda}_i,\xi)\|_2,\ \sup_{\boldsymbol{y}_j,\boldsymbol{\lambda}_j,\xi} \left\|\partial_{\boldsymbol{y}_j} h_j(\boldsymbol{y}_j,\boldsymbol{\lambda}_j,\xi)\right\|_2\right\} \le \frac{\eta_A}{2\max\left\{n, \sum_{i,j} \frac{\bar{p}_{ji}}{\bar{\delta}_{i\to j}}\right\}},$$

$$\max\left\{\sup_{\boldsymbol{y}_i,\boldsymbol{\lambda}_i,\xi} \|\partial_{\boldsymbol{\lambda}_i} g_i(\boldsymbol{y}_i,\boldsymbol{\lambda}_i,\xi)\|_2,\ \sup_{\boldsymbol{y}_j,\boldsymbol{\lambda}_j,\xi} \left\|\partial_{\boldsymbol{\lambda}_j} h_j(\boldsymbol{y}_j,\boldsymbol{\lambda}_j,\xi)\right\|_2\right\} \le \frac{\eta_B}{2\max\left\{n, \sum_{i,j} \frac{\bar{p}_{ji}}{\bar{\delta}_{i\to j}}\right\}}.$$

Under Assumption 8, we have

$$\begin{aligned}
\|\boldsymbol{A}\|_2 &\le \sum_i \sup_{\boldsymbol{y}_i,\boldsymbol{\lambda}_i,\xi} \|\partial_{\boldsymbol{y}_i} g_i(\boldsymbol{y}_i,\boldsymbol{\lambda}_i,\xi)\|_2 + \sum_{i,j} \bar{p}_{ji} \sup_{\boldsymbol{y}_j,\boldsymbol{\lambda}_j,\xi} \left\|\partial_{\boldsymbol{y}_j} h_j(\boldsymbol{y}_j,\boldsymbol{\lambda}_i,\xi)\right\|_2 \\
&\le \left(n + \sum_{i,j} \bar{p}_{ji}\right) \frac{\eta_A}{2\max\left\{n, \sum_{i,j} \frac{\bar{p}_{ji}}{\bar{\delta}_{i\to j}}\right\}} \le \eta_A, \\
\left\|\hat{\boldsymbol{A}}\right\|_2 &\le \sum_i \sup_{\boldsymbol{y}_i,\boldsymbol{\lambda}_i,\xi} \|\partial_{\boldsymbol{y}_i} g_i(\boldsymbol{y}_i,\boldsymbol{\lambda}_i,\xi)\|_2 + \sum_{i,j} \frac{\bar{p}_{ji}}{\bar{\delta}_{i\to j}} \sup_{\boldsymbol{y}_j,\boldsymbol{\lambda}_j,\xi} \left\|\partial_{\boldsymbol{y}_j} h_j(\boldsymbol{y}_j,\boldsymbol{\lambda}_i,\xi)\right\|_2 \\
&\le \left(n + \sum_{i,j} \frac{\bar{p}_{ji}}{\bar{\delta}_{i\to j}}\right) \frac{\eta_A}{2\max\left\{n, \sum_{i,j} \frac{\bar{p}_{ji}}{\bar{\delta}_{i\to j}}\right\}} \le \eta_A, \\
\|\boldsymbol{B}\|_2 &\le \sum_i \sup_{\boldsymbol{y}_i,\boldsymbol{\lambda}_i,\xi} \|\partial_{\boldsymbol{\lambda}_i} g_i(\boldsymbol{y}_i,\boldsymbol{\lambda}_i,\xi)\|_2 + \sum_{i,j} \bar{p}_{ji} \sup_{\boldsymbol{y}_j,\boldsymbol{\lambda}_j,\xi} \left\|\partial_{\boldsymbol{\lambda}_j} h_j(\boldsymbol{y}_j,\boldsymbol{\lambda}_i,\xi)\right\|_2 \\
&\le \left(n + \sum_{i,j} \bar{p}_{ji}\right) \frac{\eta_B}{2\max\left\{n, \sum_{i,j} \frac{\bar{p}_{ji}}{\bar{\delta}_{i\to j}}\right\}} \le \eta_B, \\
\left\|\hat{\boldsymbol{B}}\right\|_2 &\le \sum_i \sup_{\boldsymbol{y}_i,\boldsymbol{\lambda}_i,\xi} \|\partial_{\boldsymbol{\lambda}_i} g_i(\boldsymbol{y}_i,\boldsymbol{\lambda}_i,\xi)\|_2 + \sum_{i,j} \frac{\bar{p}_{ji}}{\bar{\delta}_{i\to j}} \sup_{\boldsymbol{\lambda}_j,\boldsymbol{\lambda}_j,\xi} \left\|\partial_{\boldsymbol{y}_j} h_j(\boldsymbol{y}_j,\boldsymbol{\lambda}_i,\xi)\right\|_2 \\
&\le \left(n + \sum_{i,j} \frac{\bar{p}_{ji}}{\bar{\delta}_{i\to j}}\right) \frac{\eta_B}{2\max\left\{n, \sum_{i,j} \frac{\bar{p}_{ji}}{\bar{\delta}_{i\to j}}\right\}} \le \eta_B.
\end{aligned}$$

C.1 PRELIMINARY LEMMAS

In this section, we present a few preliminary lemmas we use in the proof of the theorems.

We recall that we can express the general HGP using the concatenated vectors and matrices as

$$\boldsymbol{v}^{(m+1)} = \alpha\left(\boldsymbol{v}^{(m)} + \hat{\boldsymbol{B}}^{(2m)}\boldsymbol{u}^{(m)}\right) + (1-\alpha)\hat{\boldsymbol{B}}^{(2m)}\boldsymbol{w}^{(m)}, \tag{16}$$

$$\boldsymbol{u}^{(m+1)} = \hat{\boldsymbol{A}}^{(2m+1)}\boldsymbol{u}^{(m)}, \tag{17}$$

$$\boldsymbol{w}^{(m+1)} = \beta\left(\hat{\boldsymbol{A}}^{(2m+1)}\boldsymbol{w}^{(m)} + \boldsymbol{c}^{\boldsymbol{y}}\right) + (1-\beta)\left(\boldsymbol{w}^{(m)} + \boldsymbol{u}^{(m+1)}\right). \tag{18}$$

with the initial conditions $\boldsymbol{v}^{(0)} \leftarrow 0$, $\boldsymbol{u}^{(0)} \leftarrow \boldsymbol{c}^{\boldsymbol{y}}$, and $\boldsymbol{w}^{(0)} \leftarrow \boldsymbol{c}^{\boldsymbol{y}}$.

The following lemmas show explicit formula of $\boldsymbol{v}$ and $\boldsymbol{w}$ and their decomposition.

Lemma 2 (Explicit Formula of $\boldsymbol{w}$)**.**

$$\begin{aligned}\boldsymbol{w}^{(M)} = &\prod_{m=0}^{M-1}\left((1-\beta)\boldsymbol{I} + \beta\hat{\boldsymbol{A}}^{(2m+1)}\right)\boldsymbol{c}^{\boldsymbol{y}} \\ &+ \sum_{i=1}^{M}\left[\prod_{m=i}^{M-1}\left((1-\beta)\boldsymbol{I} + \beta\hat{\boldsymbol{A}}^{(2m+1)}\right)\right]\left(\beta\boldsymbol{I} + (1-\beta)\left[\prod_{m=0}^{i-1}\hat{\boldsymbol{A}}^{(2m+1)}\right]\right)\boldsymbol{c}^{\boldsymbol{y}},\end{aligned} \tag{19}$$

where we define $\prod_{m\in\emptyset}(\cdot)_m = 1$ so that $\prod_{m=M}^{M-1}(\cdot)_m = \boldsymbol{I}$.

Proof. We prove the claim by induction. We first recall that

$$\boldsymbol{u}^{(M)} = \prod_{m=0}^{M-1}\hat{\boldsymbol{A}}^{(2m+1)}\boldsymbol{c}^{\boldsymbol{y}}. \tag{20}$$

By setting $m = 0$ in (18), we have

$$\begin{aligned}\boldsymbol{w}^{(1)} &= \beta\left(\hat{\boldsymbol{A}}^{(1)}\boldsymbol{w}^{(0)} + \boldsymbol{c}^{\boldsymbol{y}}\right) + (1-\beta)\left(\boldsymbol{w}^{(0)} + \boldsymbol{u}^{(1)}\right) \\ &= \beta\left(\hat{\boldsymbol{A}}^{(1)}\boldsymbol{c}^{\boldsymbol{y}} + \boldsymbol{c}^{\boldsymbol{y}}\right) + (1-\beta)\left(\boldsymbol{c}^{\boldsymbol{y}} + \hat{\boldsymbol{A}}^{(1)}\boldsymbol{c}^{\boldsymbol{y}}\right) \\ &= \boldsymbol{c}^{\boldsymbol{y}} + \hat{\boldsymbol{A}}^{(1)}\boldsymbol{c}^{\boldsymbol{y}}.\end{aligned}$$

By setting $M = 1$ in (19), we also have

$$\boldsymbol{w}^{(1)} = \left((1-\beta)\boldsymbol{I} + \beta\hat{\boldsymbol{A}}^{(1)}\right)\boldsymbol{c}^{\boldsymbol{y}} + \left(\beta\boldsymbol{I} + (1-\beta)\hat{\boldsymbol{A}}^{(1)}\right)\boldsymbol{c}^{\boldsymbol{y}} = \boldsymbol{c}^{\boldsymbol{y}} + \hat{\boldsymbol{A}}^{(1)}\boldsymbol{c}^{\boldsymbol{y}},$$

which confirms that (19) is valid when $M = 1$.

Now, suppose that the statement is true for some $M \geq 1$. Then, by (18),

$$\begin{aligned}\boldsymbol{w}^{(M+1)} &= \beta\left(\hat{\boldsymbol{A}}^{(2M+1)}w^{(M)} + C^{\boldsymbol{y}}\right) + (1-\beta)\left(\boldsymbol{w}^{(M)} + \boldsymbol{u}^{(M+1)}\right) \\ &= \beta\boldsymbol{c}^{\boldsymbol{y}} + (1-\beta)\left[\prod_{m=0}^{M}\hat{\boldsymbol{A}}^{(2m+1)}\right]\boldsymbol{c}^{\boldsymbol{y}} + \left((1-\beta)\boldsymbol{I} + \beta\hat{\boldsymbol{A}}^{(2M+1)}\right)\boldsymbol{w}^{(M)}\end{aligned}$$

$$
\begin{aligned}
&= \beta \boldsymbol{c}^{\boldsymbol{y}} + (1-\beta)\left[\prod_{m=0}^{M} \hat{\boldsymbol{A}}^{(2m+1)}\right]\boldsymbol{c}^{\boldsymbol{y}} \\
&\quad + \left((1-\beta)\boldsymbol{I} + \beta\hat{\boldsymbol{A}}^{(2M+1)}\right)\left[\prod_{m=0}^{M-1}\left((1-\beta)\boldsymbol{I} + \beta\hat{\boldsymbol{A}}^{(2m+1)}\right)\right]\boldsymbol{c}^{\boldsymbol{y}} \\
&\quad + \left((1-\beta)\boldsymbol{I} + \beta\hat{\boldsymbol{A}}^{(2M+1)}\right)\sum_{i=1}^{M}\left[\prod_{m=i}^{M-1}\left((1-\beta)\boldsymbol{I} + \beta\hat{\boldsymbol{A}}^{(2m+1)}\right)\right]\left(\beta\boldsymbol{I} + (1-\beta)\left[\prod_{m=0}^{i-1}\hat{\boldsymbol{A}}^{(2m+1)}\right]\right)\boldsymbol{c}^{\boldsymbol{y}} \\
&= \left[\prod_{m=0}^{M}\left((1-\beta)\boldsymbol{I} + \beta\hat{\boldsymbol{A}}^{(2m+1)}\right)\right]\boldsymbol{c}^{\boldsymbol{y}} \\
&\quad + 1 \times \left(\beta\boldsymbol{I} + (1-\beta)\left[\prod_{m=0}^{M}\hat{\boldsymbol{A}}^{(2m+1)}\right]\right)\boldsymbol{c}^{\boldsymbol{y}} \\
&\quad + \sum_{i=1}^{M}\left[\prod_{m=i}^{M}\left((1-\beta)\boldsymbol{I} + \beta\hat{\boldsymbol{A}}^{(2m+1)}\right)\right]\left(\beta\boldsymbol{I} + (1-\beta)\prod_{m=0}^{i-1}\hat{\boldsymbol{A}}^{(2m+1)}\right)\boldsymbol{c}^{\boldsymbol{y}} \\
&= \left[\prod_{m=0}^{M}\left((1-\beta)\boldsymbol{I} + \beta\hat{\boldsymbol{A}}^{(2m+1)}\right)\right]\boldsymbol{c}^{\boldsymbol{y}} \\
&\quad + \sum_{i=1}^{M+1}\left[\prod_{m=i}^{M}\left((1-\beta)\boldsymbol{I} + \beta\hat{\boldsymbol{A}}^{(2m+1)}\right)\right]\left(\beta\boldsymbol{I} + (1-\beta)\prod_{m=0}^{i-1}\hat{\boldsymbol{A}}^{(2m+1)}\right)\boldsymbol{c}^{\boldsymbol{y}},
\end{aligned}
$$

where the last line follows from the fact that $\prod_{i=M+1}^{M}(\cdot)_m = \boldsymbol{I}$. □

Lemma 3 (Decomposition of $\boldsymbol{w}$)**.**

$$
\begin{aligned}
&\boldsymbol{w}^{(M)} - \sum_{i=0}^{M}\boldsymbol{A}^i\boldsymbol{c}^{\boldsymbol{y}} \\
&= \left(\sum_{i=0}^{M-1}\hat{L}_1^{(i,M)}(\hat{\boldsymbol{A}}^{(2i+1)} - \boldsymbol{A})\boldsymbol{R}_1^{(i)} + \hat{\boldsymbol{L}}_2^{(i,M)}(\hat{\boldsymbol{A}}^{(2i+1)} - \boldsymbol{A})\boldsymbol{A}^i\right)\boldsymbol{c}^{\boldsymbol{y}},
\end{aligned} \tag{21}
$$

where

$$
\begin{aligned}
\hat{\boldsymbol{L}}_1^{(i,M)} &= \beta\left[\prod_{m=i+1}^{M-1}\left((1-\beta)\boldsymbol{I} + \beta\hat{\boldsymbol{A}}^{(2m+1)}\right)\right], \\
\hat{\boldsymbol{L}}_2^{(i,M)} &= (1-\beta)\sum_{j=i+1}^{M}\left[\prod_{m=j}^{M-1}\left((1-\beta)\boldsymbol{I} + \beta\hat{\boldsymbol{A}}^{(2m+1)}\right)\right]\left[\prod_{m=i+1}^{j-1}\hat{\boldsymbol{A}}^{(2m+1)}\right], \\
\boldsymbol{R}_1^{(i)} &= ((1-\beta)\boldsymbol{I} + \beta\boldsymbol{A})^i + \sum_{j=1}^{i}((1-\beta)\boldsymbol{I} + \beta\boldsymbol{A})^{i-j}\left(\beta\boldsymbol{I} + (1-\beta)\boldsymbol{A}^j\right).
\end{aligned}
$$

Proof. We first recall that, as the corollary of Lemma 2,

$$
\sum_{i=0}^{M}\boldsymbol{A}^i\boldsymbol{c}^{\boldsymbol{y}} = ((1-\beta)\boldsymbol{I} + \beta\boldsymbol{A})^M\boldsymbol{c}^{\boldsymbol{y}} + \sum_{i=1}^{M}((1-\beta)\boldsymbol{I} + \beta\boldsymbol{A})^{M-i}\left(\beta\boldsymbol{I} + (1-\beta)\boldsymbol{A}^i\right)\boldsymbol{c}^{\boldsymbol{y}}.
$$

By using Lemma 2, we can expand the difference as

$$
\begin{aligned}
&\boldsymbol{w}^{(M)} - \sum_{i=0}^{M} \boldsymbol{A}^i \boldsymbol{c}^{\boldsymbol{y}} \\
&= \left(\prod_{m=0}^{M-1} \left((1-\beta)\boldsymbol{I} + \beta \hat{\boldsymbol{A}}^{(2m+1)} \right) - \left((1-\beta)\boldsymbol{I} + \beta \boldsymbol{A} \right)^M \right) \boldsymbol{c}^{\boldsymbol{y}} \\
&\quad + \sum_{i=1}^{M} \left(\left[\prod_{m=i}^{M-1} \left((1-\beta)\boldsymbol{I} + \beta \hat{\boldsymbol{A}}^{(2m+1)} \right) \right] - \left((1-\beta)\boldsymbol{I} + \beta \boldsymbol{A} \right)^{M-i} \right) \left(\beta \boldsymbol{I} + (1-\beta)\boldsymbol{A}^i \right) \boldsymbol{c}^{\boldsymbol{y}} \\
&\quad + \sum_{i=1}^{M} \left[\prod_{m=i}^{M-1} \left((1-\beta)\boldsymbol{I} + \beta \hat{\boldsymbol{A}}^{(2m+1)} \right) \right] (1-\beta) \left(\prod_{m=0}^{i-1} \hat{\boldsymbol{A}}^{(2m+1)} - \boldsymbol{A}^i \right) \boldsymbol{c}^{\boldsymbol{y}} \\
&= \sum_{i=0}^{M-1} \left[\prod_{m=i+1}^{M-1} \left((1-\beta)\boldsymbol{I} + \beta \hat{\boldsymbol{A}}^{(2m+1)} \right) \right] \beta \left(\hat{\boldsymbol{A}}^{(2i+1)} - \boldsymbol{A} \right) \left((1-\beta)\boldsymbol{I} + \beta \boldsymbol{A} \right)^i \boldsymbol{c}^{\boldsymbol{y}} \\
&\quad + \sum_{j=1}^{M-1} \sum_{i=j}^{M-1} \left[\prod_{m=i+1}^{M-1} \left((1-\beta)\boldsymbol{I} + \beta \hat{\boldsymbol{A}}^{(2m+1)} \right) \right] \beta \left(\hat{\boldsymbol{A}}^{(2i+1)} - \boldsymbol{A} \right) \left((1-\beta)\boldsymbol{I} + \beta \boldsymbol{A} \right)^{i-j} \\
&\qquad \times \left(\beta \boldsymbol{I} + (1-\beta)\boldsymbol{A}^j \right) \boldsymbol{c}^{\boldsymbol{y}} \\
&\quad + \sum_{j=1}^{M} \left[\prod_{m=j}^{M-1} \left((1-\beta)\boldsymbol{I} + \beta \hat{\boldsymbol{A}}^{(2m+1)} \right) \right] (1-\beta) \sum_{i=0}^{j-1} \left[\prod_{m=i+1}^{j-1} \hat{\boldsymbol{A}}^{(2m+1)} \right] \left(\hat{\boldsymbol{A}}^{(2i+1)} - \boldsymbol{A} \right) \boldsymbol{A}^i \boldsymbol{c}^{\boldsymbol{y}} \\
&= \sum_{i=0}^{M-1} \left[\prod_{m=i+1}^{M-1} \left((1-\beta)\boldsymbol{I} + \beta \hat{\boldsymbol{A}}^{(2m+1)} \right) \right] \beta \left(\hat{\boldsymbol{A}}^{(2i+1)} - \boldsymbol{A} \right) \left((1-\beta)\boldsymbol{I} + \beta \boldsymbol{A} \right)^i \boldsymbol{c}^{\boldsymbol{y}} \\
&\quad + \sum_{i=1}^{M-1} \left[\prod_{m=i+1}^{M-1} \left((1-\beta)\boldsymbol{I} + \beta \hat{\boldsymbol{A}}^{(2m+1)} \right) \right] \beta \left(\hat{\boldsymbol{A}}^{(2i+1)} - \boldsymbol{A} \right) \\
&\qquad \times \sum_{j=1}^{i} \left((1-\beta)\boldsymbol{I} + \beta \boldsymbol{A} \right)^{i-j} \left(\beta \boldsymbol{I} + (1-\beta)\boldsymbol{A}^j \right) \boldsymbol{c}^{\boldsymbol{y}} \\
&\quad + \sum_{i=0}^{M-1} \sum_{j=i+1}^{M} \left[\prod_{m=j}^{M-1} \left((1-\beta)\boldsymbol{I} + \beta \hat{\boldsymbol{A}}^{(2m+1)} \right) \right] (1-\beta) \left[\prod_{m=i+1}^{j-1} \hat{\boldsymbol{A}}^{(2m+1)} \right] \left(\hat{\boldsymbol{A}}^{(2i+1)} - \boldsymbol{A} \right) \boldsymbol{A}^i \boldsymbol{c}^{\boldsymbol{y}} \\
&= \sum_{i=0}^{M-1} \underbrace{\beta \left[\prod_{m=i+1}^{M-1} \left((1-\beta)\boldsymbol{I} + \beta \hat{\boldsymbol{A}}^{(2m+1)} \right) \right]}_{=\hat{\boldsymbol{L}}_1^{(i,M)}} \left(\hat{\boldsymbol{A}}^{(2i+1)} - \boldsymbol{A} \right) \\
&\qquad \times \underbrace{\left(\left((1-\beta)\boldsymbol{I} + \beta \boldsymbol{A} \right)^i + \sum_{j=1}^{i} \left((1-\beta)\boldsymbol{I} + \beta \boldsymbol{A} \right)^{i-j} \left(\beta \boldsymbol{I} + (1-\beta)\boldsymbol{A}^j \right) \right)}_{=\boldsymbol{R}_1^{(i)}} \boldsymbol{c}^{\boldsymbol{y}} \\
&\quad + \sum_{i=0}^{M-1} \underbrace{(1-\beta) \sum_{j=i+1}^{M} \left[\prod_{m=j}^{M-1} \left((1-\beta)\boldsymbol{I} + \beta \hat{\boldsymbol{A}}^{(2m+1)} \right) \right] \left[\prod_{m=i+1}^{j-1} \hat{\boldsymbol{A}}^{(2m+1)} \right]}_{=\hat{\boldsymbol{L}}_2^{(i,M)}} \left(\hat{\boldsymbol{A}}^{(2i+1)} - \boldsymbol{A} \right) \boldsymbol{A}^i \boldsymbol{c}^{\boldsymbol{y}}.
\end{aligned}
$$

□

Lemma 4 (Explicit Formula of $\boldsymbol{v}$)**.**

$$\boldsymbol{v}^{(M+1)} = \sum_{i=0}^{M} \alpha^{M-i+1} \hat{\boldsymbol{B}}^{(2i)} \boldsymbol{u}^{(i)} + (1-\alpha) \sum_{i=0}^{M} \alpha^{M-i} \hat{\boldsymbol{B}}^{(2i)} \boldsymbol{w}^{(i)}. \tag{22}$$

Proof. We prove the claim by induction. By setting $m = 0$ in (16), we have

$$\begin{aligned}\boldsymbol{v}^{(1)} &= \alpha \left(\boldsymbol{v}^{(0)} + \hat{\boldsymbol{B}}^{(0)} \boldsymbol{u}^{(0)} \right) + (1-\alpha) \hat{\boldsymbol{B}}^{(0)} \boldsymbol{w}^{(0)} \\ &= \alpha \hat{\boldsymbol{B}}^{(0)} \boldsymbol{c}^{\boldsymbol{y}} + (1-\alpha) \hat{\boldsymbol{B}}^{(0)} \boldsymbol{c}^{\boldsymbol{y}} = \hat{\boldsymbol{B}}^{(0)} \boldsymbol{c}^{\boldsymbol{y}}.\end{aligned}$$

By setting $M = 0$ in (22), we also have

$$\boldsymbol{v}^{(1)} = \alpha \hat{\boldsymbol{B}}^{(0)} \boldsymbol{c}^{\boldsymbol{y}} + (1-\alpha) \alpha \hat{\boldsymbol{B}}^{(0)} \boldsymbol{w}^{(0)} = \hat{\boldsymbol{B}}^{(0)} \boldsymbol{c}^{\boldsymbol{y}},$$

which confirms that (22) is valid when $M = 0$.

Now, suppose that the statement is true for some $M \geq 1$. Then, by (16),

$$\begin{aligned}\boldsymbol{v}^{(M+1)} &= \alpha \left(\boldsymbol{v}^{(M)} + \hat{\boldsymbol{B}}^{(2M)} \boldsymbol{u}^{(M)} \right) + (1-\alpha) \hat{\boldsymbol{B}}^{(2M)} \boldsymbol{w}^{(M)} \\ &= \alpha \left(\sum_{i=0}^{M-1} \alpha^{M-i} \hat{\boldsymbol{B}}^{(2i)} \boldsymbol{u}^{(i)} + (1-\alpha) \sum_{i=0}^{M-1} \alpha^{M-i-1} \hat{\boldsymbol{B}}^{(2i)} \boldsymbol{w}^{(i)} \right) \\ &\quad + \alpha \hat{\boldsymbol{B}}^{(2M)} \boldsymbol{u}^{(M)} + (1-\alpha) \hat{\boldsymbol{B}}^{(2M)} \boldsymbol{w}^{(M)} \\ &= \left(\sum_{i=0}^{M-1} \alpha^{M-i+1} \hat{\boldsymbol{B}}^{(2i)} \boldsymbol{u}^{(i)} + \alpha \hat{\boldsymbol{B}}^{(2M)} \boldsymbol{u}^{(M)} \right) \\ &\quad + (1-\alpha) \left(\sum_{i=0}^{M-1} \alpha^{M-i} \hat{\boldsymbol{B}}^{(2i)} \boldsymbol{w}^{(i)} + \hat{\boldsymbol{B}}^{(2M)} \boldsymbol{w}^{(M+1)} \right) \\ &= \sum_{i=0}^{M} \alpha^{M-i+1} \hat{\boldsymbol{B}}^{(2i)} \boldsymbol{u}^{(i)} + (1-\alpha) \sum_{i=0}^{M} \alpha^{M-i} \hat{\boldsymbol{B}}^{(2i)} \boldsymbol{w}^{(i)}.\end{aligned}$$

□

Lemma 5 (Decomposition of $\boldsymbol{v}$)**.**

$$\begin{aligned}&\boldsymbol{v}^{(M+1)} - \boldsymbol{B} \sum_{i=0}^{M} \boldsymbol{A}^i \boldsymbol{c}^{\boldsymbol{y}} \\ &= \sum_{i=0}^{M} (\hat{\boldsymbol{B}}^{(2i)} - \boldsymbol{B}) \boldsymbol{R}_3^{(i,M)} \boldsymbol{c}^{\boldsymbol{y}} + \sum_{i=0}^{M-1} \left(\hat{\boldsymbol{L}}_4^{(i,M)} (\hat{\boldsymbol{A}}^{(2i+1)} - \boldsymbol{A}) \boldsymbol{A}^i + \hat{\boldsymbol{L}}_5^{(i,M)} (\hat{\boldsymbol{A}}^{(2i+1)} - \boldsymbol{A}) \boldsymbol{R}_1^{(i)} \right) \boldsymbol{c}^{\boldsymbol{y}},\end{aligned} \tag{23}$$

where

$$\begin{aligned}\boldsymbol{R}_3^{(i,M)} &= \alpha^{M-i+1} \boldsymbol{A}^i + (1-\alpha) \alpha^{M-i} \sum_{j=0}^{i} \boldsymbol{A}^j, \\ \hat{\boldsymbol{L}}_4^{(i,M)} &= \sum_{j=i+1}^{M} \alpha^{M-j+1} \hat{\boldsymbol{B}}^{(2j)} \left[\prod_{m=i+1}^{j-1} \hat{\boldsymbol{A}}^{(2m+1)} \right] \\ &\quad + (1-\alpha)(1-\beta) \sum_{j=i+1}^{M} \alpha^{M-j} \hat{\boldsymbol{B}}^{(2j)} \sum_{k=i+1}^{j} \left[\prod_{m=k}^{j-1} \left((1-\beta) \boldsymbol{I} + \beta \hat{\boldsymbol{A}}^{(2m+1)} \right) \right] \left[\prod_{m=i+1}^{k-1} \hat{\boldsymbol{A}}^{(2m+1)} \right], \\ \hat{\boldsymbol{L}}_5^{(i,M)} &= (1-\alpha) \beta \sum_{j=i+1}^{M} \alpha^{M-j} \hat{\boldsymbol{B}}^{(2j)} \left[\prod_{m=i+1}^{j-1} \left((1-\beta) \boldsymbol{I} + \beta \hat{\boldsymbol{A}}^{(2m+1)} \right) \right].\end{aligned}$$

Proof. We first recall that, as the corollary of Lemma 4,

$$\boldsymbol{B}\sum_{i=0}^{M}\boldsymbol{A}^i\boldsymbol{c}^{\boldsymbol{y}} = \sum_{i=0}^{M}\alpha^{M-i+1}\boldsymbol{B}\boldsymbol{A}^i\boldsymbol{c}^{\boldsymbol{y}} + (1-\alpha)\sum_{i=0}^{M}\alpha^{M-i}\boldsymbol{B}\sum_{j=0}^{i}\boldsymbol{A}^j\boldsymbol{c}^{\boldsymbol{y}}.$$

By using Lemma 4 and Lemma 2, we can expand the difference as

$$\begin{aligned}
&\boldsymbol{v}^{(M+1)} - \boldsymbol{B}\sum_{i=0}^{M}\boldsymbol{A}^i\boldsymbol{c}^{\boldsymbol{y}} \\
&= \sum_{i=0}^{M}\alpha^{M-i+1}\left(\hat{\boldsymbol{B}}^{(2i)}\left[\prod_{m=0}^{i-1}\hat{\boldsymbol{A}}^{(2m+1)}\right] - \boldsymbol{B}\boldsymbol{A}^i\right)\boldsymbol{c}^{\boldsymbol{y}} + (1-\alpha)\sum_{i=0}^{M}\alpha^{M-i}\left(\hat{\boldsymbol{B}}^{(2i)}\boldsymbol{w}^{(i)} - \boldsymbol{B}\sum_{j=0}^{i}\boldsymbol{A}^j\boldsymbol{c}^{\boldsymbol{y}}\right) \\
&= \sum_{i=0}^{M}\alpha^{M-i+1}\left(\hat{\boldsymbol{B}}^{(2i)} - \boldsymbol{B}\right)\boldsymbol{A}^i\boldsymbol{c}^{\boldsymbol{y}} + \sum_{j=1}^{M}\alpha^{M-j+1}\hat{\boldsymbol{B}}^{(2j)}\sum_{i=0}^{j-1}\left[\prod_{m=i+1}^{j-1}\hat{\boldsymbol{A}}^{(2m+1)}\right]\left(\hat{\boldsymbol{A}}^{(2i+1)} - \boldsymbol{A}\right)\boldsymbol{A}^i\boldsymbol{c}^{\boldsymbol{y}} \\
&\quad + (1-\alpha)\sum_{i=0}^{M}\alpha^{M-i}\left(\hat{\boldsymbol{B}}^{(2i)} - \boldsymbol{B}\right)\sum_{j=0}^{i}\boldsymbol{A}^j\boldsymbol{c}^{\boldsymbol{y}} + (1-\alpha)\sum_{i=1}^{M}\alpha^{M-i}\hat{\boldsymbol{B}}^{(2i)}\left(\boldsymbol{w}^{(i)} - \sum_{j=0}^{i}\boldsymbol{A}^j\boldsymbol{c}^{\boldsymbol{y}}\right).
\end{aligned}$$

By substituting (21), we have

$$\begin{aligned}
&\boldsymbol{v}^{(M+1)} - \boldsymbol{B}\sum_{i=0}^{M}\boldsymbol{A}^i\boldsymbol{c}^{\boldsymbol{y}} \\
&= \sum_{i=0}^{M}\alpha^{M-i+1}\left(\hat{\boldsymbol{B}}^{(2i)} - \boldsymbol{B}\right)\boldsymbol{A}^i\boldsymbol{c}^{\boldsymbol{y}} + \sum_{j=1}^{M}\alpha^{M-j+1}\hat{\boldsymbol{B}}^{(2j)}\sum_{i=0}^{j-1}\left[\prod_{m=i+1}^{j-1}\hat{\boldsymbol{A}}^{(2m+1)}\right]\left(\hat{\boldsymbol{A}}^{(2i+1)} - \boldsymbol{A}\right)\boldsymbol{A}^i\boldsymbol{c}^{\boldsymbol{y}} \\
&\quad + (1-\alpha)\sum_{i=0}^{M}\alpha^{M-i}\left(\hat{\boldsymbol{B}}^{(2i)} - \boldsymbol{B}\right)\sum_{j=0}^{i}\boldsymbol{A}^j\boldsymbol{c}^{\boldsymbol{y}} \\
&\quad + (1-\alpha)\sum_{j=1}^{M}\alpha^{M-j}\hat{\boldsymbol{B}}^{(2j)}\left(\sum_{i=0}^{j-1}\hat{\boldsymbol{L}}_1^{(i,j)}(\hat{\boldsymbol{A}}^{(2i+1)} - \boldsymbol{A})\boldsymbol{R}_1^{(i)} + \hat{\boldsymbol{L}}_2^{(i,j)}(\hat{\boldsymbol{A}}^{(2i+1)} - \boldsymbol{A})\boldsymbol{A}^i\right)\boldsymbol{c}^{\boldsymbol{y}} \\
&= \sum_{i=0}^{M}\left(\hat{\boldsymbol{B}}^{(2i)} - \boldsymbol{B}\right)\underbrace{\left(\alpha^{M-i+1}\boldsymbol{A}^i + (1-\alpha)\alpha^{M-i}\sum_{j=0}^{i}\boldsymbol{A}^j\right)}_{=\boldsymbol{R}_3^{(i,M)}}\boldsymbol{c}^{\boldsymbol{y}} \\
&\quad + \sum_{i=0}^{M-1}\underbrace{\left(\sum_{j=i+1}^{M}\alpha^{M-j+1}\hat{\boldsymbol{B}}^{(2j)}\left[\prod_{m=i+1}^{j-1}\hat{\boldsymbol{A}}^{(2m+1)}\right] + (1-\alpha)\sum_{j=i+1}^{M}\alpha^{M-j}\hat{\boldsymbol{B}}^{(2j)}\hat{\boldsymbol{L}}_2^{(i,j)}\right)}_{=\hat{\boldsymbol{L}}_4^{(i,M)}}\left(\hat{\boldsymbol{A}}^{(2i+1)} - \boldsymbol{A}\right)\boldsymbol{A}^i\boldsymbol{c}^{\boldsymbol{y}} \\
&\quad + \sum_{i=0}^{M-1}\underbrace{(1-\alpha)\sum_{j=i+1}^{M}\alpha^{M-j}\hat{\boldsymbol{B}}^{(2j)}\hat{\boldsymbol{L}}_1^{(i,j)}}_{=\hat{\boldsymbol{L}}_5^{(i,M)}}(\hat{\boldsymbol{A}}^{(2i+1)} - \boldsymbol{A})\boldsymbol{R}_1^{(i)}\boldsymbol{c}^{\boldsymbol{y}}.
\end{aligned}$$

By substituting $\boldsymbol{L}_2^{(i,j)}$, $\boldsymbol{L}_1^{(i,j)}$, we obtain the claim. □

To bound the estimation error of hyper-gradient, we need to bound each term of (23). The following lemma gives the bounds for each coefficient matrices in (23).

Lemma 6. Under Assumption 8, we have

$$\left\|\boldsymbol{R}_3^{(i,M)}\right\|_2 \leq \frac{1-\alpha}{1-\eta_A}\alpha^{M-i} + \frac{1}{1-\eta_A}\alpha^{M-i+1}\eta_A^i - \frac{1}{1-\eta_A}\alpha^{M-i}\eta_A^{i+1}, \tag{24}$$

$$\begin{aligned}\left\|\hat{\boldsymbol{L}}_4^{(i,M)}\right\|_2 \leq{}& \frac{\eta_B\alpha\beta}{\alpha-(1-\beta+\beta\eta_A)}\alpha^{M-i} \\ &- \frac{\eta_B}{1-\eta_A}\eta_A^{M-i} - \frac{\eta_B}{1-\eta_A}\frac{1-\alpha}{\alpha-(1-\beta+\beta\eta_A)}(1-\beta+\beta\eta_A)^{M-i+1},\end{aligned} \tag{25}$$

$$\left\|\hat{\boldsymbol{L}}_5^{(i,M)}\right\|_2 \leq \eta_B\frac{(1-\alpha)\beta}{\alpha-(1-\beta+\beta\eta_A)}\left(\alpha^{M-i}-(1-\beta+\beta\eta_A)^{M-i}\right), \tag{26}$$

$$\left\|\boldsymbol{R}_1^{(i)}\right\|_2 \leq \frac{1-\eta_A^{i+1}}{1-\eta_A}. \tag{27}$$

Proof. Recall that Assumption 8 ensures $\|\boldsymbol{A}\|_2 \leq \eta_A$, $\left\|\hat{\boldsymbol{A}}\right\|_2 \leq \eta_A$, $\|\boldsymbol{B}\|_2 \leq \eta_B$, $\left\|\hat{\boldsymbol{B}}\right\|_2 \leq \eta_B$. Then, we have

$$\begin{aligned}\left\|\boldsymbol{R}_3^{(i,M)}\right\|_2 &\leq \alpha^{M-i+1}\|\boldsymbol{A}\|_2^i + (1-\alpha)\alpha^{M-i}\sum_{j=0}^{i}\|\boldsymbol{A}\|_2^j \\ &\leq \alpha^{M-i+1}\eta_A^i + (1-\alpha)\alpha^{M-i}\sum_{j=0}^{i}\eta_A^j \\ &= \alpha^{M-i+1}\eta_A^i + (1-\alpha)\alpha^{M-i}\frac{1-\eta_A^{i+1}}{1-\eta_A} \\ &= \alpha^{M-i+1}\eta_A^i + \frac{1-\alpha}{1-\eta_A}\alpha^{M-i} - \frac{1}{1-\eta_A}\alpha^{M-i}\eta_A^{i+1} + \frac{\eta_A}{1-\eta_A}\alpha^{M-i+1}\eta_A^i \\ &= \frac{1-\alpha}{1-\eta_A}\alpha^{M-i} + \frac{1}{1-\eta_A}\alpha^{M-i+1}\eta_A^i - \frac{1}{1-\eta_A}\alpha^{M-i}\eta_A^{i+1},\end{aligned}$$

$$\begin{aligned}\left\|\hat{\boldsymbol{L}}_4^{(i,M)}\right\|_2 \leq{}& \sum_{j=i+1}^{M}\alpha^{M-j+1}\left\|\hat{\boldsymbol{B}}^{(2j)}\right\|_2\prod_{m=i+1}^{j-1}\left\|\hat{\boldsymbol{A}}^{(2m+1)}\right\|_2 \\ &+ (1-\alpha)(1-\beta)\sum_{j=i+1}^{M}\alpha^{M-j}\left\|\hat{\boldsymbol{B}}^{(2j)}\right\|_2\sum_{k=i+1}^{j}\prod_{m=k}^{j-1}\left\|(1-\beta)I+\beta\hat{\boldsymbol{A}}^{(2m+1)}\right\|_2\prod_{m=i+1}^{k-1}\left\|\hat{\boldsymbol{A}}^{(2m+1)}\right\|_2 \\ \leq{}& \eta_B\sum_{j=i+1}^{M}\alpha^{M-j+1}\eta_A^{j-i-1} + \eta_B(1-\alpha)(1-\beta)\sum_{j=i+1}^{M}\alpha^{M-j}\sum_{k=i+1}^{j}(1-\beta+\beta\eta_A)^{j-k}\eta_A^{k-i-1} \\ ={}& \eta_B\frac{\alpha}{\alpha-\eta_A}\left(\alpha^{M-i}-\eta_A^{M-i}\right) + \eta_B\frac{1-\alpha}{1-\eta_A}\sum_{j=i+1}^{M}\alpha^{M-j}\left((1-\beta+\beta\eta_A)^{j-i}-\eta_A^{j-i}\right) \\ ={}& \eta_B\frac{\alpha}{\alpha-\eta_A}\left(\alpha^{M-i}-\eta_A^{M-i}\right) \\ &+ \eta_B\frac{1-\alpha}{1-\eta_A}\left(\frac{(1-\beta+\beta\eta_A)\left(\alpha^{M-i}-(1-\beta+\beta\eta_A)^{M-i}\right)}{\alpha-(1-\beta+\beta\eta_A)} - \frac{\eta_A\left(\alpha^{M-i}-\eta_A^{M-i}\right)}{\alpha-\eta_A}\right) \\ ={}& \eta_B\frac{1}{1-\eta_A}\left(\alpha^{M-i}-\eta_A^{M-i}\right) + \eta_B\frac{1-\alpha}{1-\eta_A}\frac{1-\beta+\beta\eta_A}{\alpha-(1-\beta+\beta\eta_A)}\left(\alpha^{M-i}-(1-\beta+\beta\eta_A)^{M-i}\right) \\ ={}& \frac{\eta_B\alpha\beta}{\alpha-(1-\beta+\beta\eta_A)}\alpha^{M-i} - \frac{\eta_B}{1-\eta_A}\eta_A^{M-i} - \frac{\eta_B}{1-\eta_A}\frac{1-\alpha}{\alpha-(1-\beta+\beta\eta_A)}(1-\beta+\beta\eta_A)^{M-i+1},\end{aligned}$$

$$
\begin{aligned}
\left\|\hat{\boldsymbol{L}}_5^{(i,M)}\right\|_2 &\leq (1-\alpha)\beta \sum_{j=i+1}^{M} \alpha^{M-j} \left\|\hat{\boldsymbol{B}}^{(2j)}\right\|_2 \prod_{m=i+1}^{j-1} \left\|(1-\beta)\boldsymbol{I} + \beta \hat{\boldsymbol{A}}^{(2m+1)}\right\|_2 \\
&\leq \eta_B (1-\alpha)\beta \sum_{j=i+1}^{M} \alpha^{M-j} (1-\beta+\beta\eta_A)^{j-i-1} \\
&= \eta_B (1-\alpha)\beta \frac{\alpha^{M-i} - (1-\beta+\beta\eta_A)^{M-i}}{\alpha - (1-\beta+\beta\eta_A)}, \\
\left\|\boldsymbol{R}_1^{(i)}\right\|_2 &\leq \|(1-\beta)\boldsymbol{I} + \beta\boldsymbol{A}\|_2^i + \sum_{j=1}^{i} \|(1-\beta)\boldsymbol{I} + \beta\boldsymbol{A}\|_2^{i-j} \left(\beta + (1-\beta)\|\boldsymbol{A}\|_2^j\right) \\
&\leq (1-\beta+\beta\eta_A)^i + \beta \sum_{j=1}^{i} (1-\beta+\beta\eta_A)^{i-j} + (1-\beta) \sum_{j=1}^{i} (1-\beta+\beta\eta_A)^{i-j} \eta_A^j \\
&= (1-\beta+\beta\eta_A)^i + \frac{1-(1-\beta+\beta\eta_A)^i}{1-\eta_A} + \eta_A \frac{(1-\beta+\beta\eta_A)^i - \eta_A^i}{1-\eta_A} \\
&= \frac{1-\eta_A^{i+1}}{1-\eta_A}.
\end{aligned}
$$

□

C.2 Decomposition of $\hat{\boldsymbol{A}}$, $\hat{\boldsymbol{B}}$

We can decompose the difference $\hat{\boldsymbol{A}} - \boldsymbol{A}$ and $\hat{\boldsymbol{B}} - \boldsymbol{B}$ as

$$
\begin{aligned}
\hat{\boldsymbol{A}} - \boldsymbol{A} &= \left[\mathbb{1}_{ji}(\hat{\boldsymbol{A}}_i^{\psi} - \boldsymbol{A}_i^{\psi}) + \bar{p}_{ji}\left(\frac{\delta_{i\to j}}{\bar{\delta}_{i\to j}}\hat{\boldsymbol{A}}_j^{\varphi} - \boldsymbol{A}_j^{\varphi}\right)\right]_{ji} \\
&= \left[\left(\frac{\delta_{i\to j}}{\bar{\delta}_{i\to j}} - 1\right)\bar{p}_{ji}\hat{\boldsymbol{A}}_j^{\varphi}\right]_{ji} + \left[\mathbb{1}_{ji}(\hat{\boldsymbol{A}}_i^{\psi} - \boldsymbol{A}_i^{\psi}) + \bar{p}_{ji}\left(\hat{\boldsymbol{A}}_j^{\varphi} - \boldsymbol{A}_j^{\varphi}\right)\right]_{ji} \\
&= \sum_{i,j=1}^{n} \boldsymbol{e}_j \boldsymbol{e}_i^{\top} \otimes \left(\frac{\delta_{i\to j}}{\bar{\delta}_{i\to j}} - 1\right)\bar{p}_{ji}\hat{\boldsymbol{A}}_j^{\varphi} \\
&\quad + \sum_{i,j=1}^{n} \boldsymbol{e}_j \boldsymbol{e}_i^{\top} \otimes \left(\frac{\mathbb{1}_{ji}}{|\zeta_i|}\sum_{\xi\in\zeta_i}\left(\partial_{\boldsymbol{y}_i} g_i(\boldsymbol{y}_i^*, \boldsymbol{\lambda}_i, \xi) - \boldsymbol{A}_i^{\psi}\right) + \frac{\bar{p}_{ji}}{|\zeta_j|}\sum_{\xi\in\zeta_j}\left(\partial_{\boldsymbol{y}_j} h_j(\boldsymbol{y}_j^*, \boldsymbol{\lambda}_j, \xi) - \boldsymbol{A}_j^{\varphi}\right)\right), \\
\hat{\boldsymbol{B}} - \boldsymbol{B} &= \left[\mathbb{1}_{ji}(\hat{\boldsymbol{B}}_i^{\psi} - \boldsymbol{B}_i^{\psi}) + \bar{p}_{ji}\left(\frac{\delta_{i\to j}}{\bar{\delta}_{i\to j}}\hat{\boldsymbol{B}}_j^{\varphi} - \boldsymbol{B}_j^{\varphi}\right)\right]_{ji} \\
&= \left[\left(\frac{\delta_{i\to j}}{\bar{\delta}_{i\to j}} - 1\right)\bar{p}_{ji}\hat{\boldsymbol{B}}_j^{\varphi}\right]_{ji} + \left[\mathbb{1}_{ji}(\hat{\boldsymbol{B}}_i^{\psi} - \boldsymbol{B}_i^{\psi}) + \bar{p}_{ji}\left(\hat{\boldsymbol{B}}_j^{\varphi} - \boldsymbol{B}_j^{\varphi}\right)\right]_{ji} \\
&= \sum_{i,j=1}^{n} \boldsymbol{e}_j \boldsymbol{e}_i^{\top} \otimes \left(\frac{\delta_{i\to j}}{\bar{\delta}_{i\to j}} - 1\right)\bar{p}_{ji}\hat{\boldsymbol{B}}_j^{\varphi} \\
&\quad + \sum_{i,j=1}^{n} \boldsymbol{e}_j \boldsymbol{e}_i^{\top} \otimes \left(\frac{\mathbb{1}_{ji}}{|\zeta_i|}\sum_{\xi\in\zeta_i}\left(\partial_{\boldsymbol{\lambda}_i} g_i(\boldsymbol{y}_i^*, \boldsymbol{\lambda}_i, \xi) - \boldsymbol{B}_i^{\psi}\right) + \frac{\bar{p}_{ji}}{|\zeta_j|}\sum_{\xi\in\zeta_j}\left(\partial_{\boldsymbol{\lambda}_j} h_j(\boldsymbol{y}_j^*, \boldsymbol{\lambda}_j, \xi) - \boldsymbol{B}_j^{\varphi}\right)\right),
\end{aligned}
$$

where $\boldsymbol{e}_i, \boldsymbol{e}_j$ are i-th and j-th canonical basis vectors.

By using these expressions, we can rewrite Lemma 5 as

$$\boldsymbol{v}^{(M+1)} - \boldsymbol{B}\sum_{i=0}^{M}\boldsymbol{A}^i\boldsymbol{c}^{\boldsymbol{y}}$$
$$= \sum_{i=0}^{M}\sum_{s,t=1}^{n}\boldsymbol{X}_{B,st}^{(i)}\boldsymbol{c}^{\boldsymbol{y}} + \sum_{i=0}^{M}\sum_{t=1}^{n}\sum_{\xi\in\zeta_s^{(2i)}}\boldsymbol{Y}_{B,t}^{(i,\xi)}\boldsymbol{c}^{\boldsymbol{y}} + \sum_{i=0}^{M-1}\sum_{s,t=1}^{n}\boldsymbol{X}_{A,st}^{(i)}\boldsymbol{c}^{\boldsymbol{y}} + \sum_{i=0}^{M-1}\sum_{t=1}^{n}\sum_{\xi\in\zeta_s^{(2i+1)}}\boldsymbol{Y}_{A,t}^{(i,\xi)}\boldsymbol{c}^{\boldsymbol{y}},$$

where

$$\boldsymbol{X}_{B,st}^{(i)} = \left(\boldsymbol{e}_t\boldsymbol{e}_s^\top \otimes \left(\frac{\delta_{s\to t}^{(2i)}}{\bar{\delta}_{s\to t}} - 1\right)\bar{p}_{ts}\hat{\boldsymbol{B}}_t^{\varphi(2i)}\right)\boldsymbol{R}_3^{(i,M)},$$
$$\boldsymbol{Y}_{B,t}^{(i)} = \sum_{s=1}^{n}\left(\boldsymbol{e}_t\boldsymbol{e}_s^\top \otimes \frac{1}{|\zeta_t^{(2i)}|}\left(\mathbb{1}_{ts}\partial_{\boldsymbol{\lambda}_t}g_t(\boldsymbol{y}_t^*,\boldsymbol{\lambda}_t,\xi) + \bar{p}_{ts}\partial_{\boldsymbol{\lambda}_t}h_t(\boldsymbol{y}_t^*,\boldsymbol{\lambda}_t,\xi) - \mathbb{1}_{ts}\boldsymbol{B}_t^{\psi} - \bar{p}_{ts}\boldsymbol{B}_t^{\varphi}\right)\right)\boldsymbol{R}_3^{(i,M)},$$
$$\boldsymbol{X}_{A,st}^{(i)} = \hat{\boldsymbol{L}}_4^{(i,M)}\left(\boldsymbol{e}_t\boldsymbol{e}_s^\top \otimes \left(\frac{\delta_{s\to t}^{(2i+1)}}{\bar{\delta}_{s\to t}} - 1\right)\bar{p}_{ts}\hat{\boldsymbol{A}}_t^{\varphi(2i+1)}\right)\boldsymbol{A}^i$$
$$+ \hat{\boldsymbol{L}}_5^{(i,M)}\left(\boldsymbol{e}_t\boldsymbol{e}_s^\top \otimes \left(\frac{\delta_{s\to t}^{(2i+1)}}{\bar{\delta}_{s\to t}} - 1\right)\bar{p}_{ts}\hat{\boldsymbol{A}}_t^{\varphi(2i+1)}\right)\boldsymbol{R}_1^{(i)},$$
$$\boldsymbol{Y}_{B,t}^{(i)} = \hat{\boldsymbol{L}}_4^{(i,M)}\sum_{s=1}^{n}\left(\boldsymbol{e}_t\boldsymbol{e}_s^\top \otimes \frac{1}{|\zeta_t^{(2i+1)}|}\left(\mathbb{1}_{ts}\partial_{\boldsymbol{y}_t}g_t(\boldsymbol{y}_t^*,\boldsymbol{\lambda}_t,\xi) + \bar{p}_{ts}\partial_{\boldsymbol{y}_t}h_t(\boldsymbol{y}_t^*,\boldsymbol{\lambda}_t,\xi) - \mathbb{1}_{ts}\boldsymbol{A}_t^{\psi} - \bar{p}_{ts}\boldsymbol{A}_t^{\varphi}\right)\right)\boldsymbol{A}^i$$
$$+ \hat{\boldsymbol{L}}_5^{(i,M)}\sum_{s=1}^{n}\left(\boldsymbol{e}_t\boldsymbol{e}_s^\top \otimes \frac{1}{|\zeta_t^{(2i+1)}|}\left(\mathbb{1}_{ts}\partial_{\boldsymbol{y}_t}g_t(\boldsymbol{y}_t^*,\boldsymbol{\lambda}_t,\xi) + \bar{p}_{ts}\partial_{\boldsymbol{y}_t}h_t(\boldsymbol{y}_t^*,\boldsymbol{\lambda}_t,\xi) - \mathbb{1}_{ts}\boldsymbol{A}_t^{\psi} - \bar{p}_{ts}\boldsymbol{A}_t^{\varphi}\right)\right)\boldsymbol{R}_1^{(i)}.$$

Here, we note that $\hat{\boldsymbol{L}}_4^{(i,M)}$ and $\hat{\boldsymbol{L}}_5^{(i,M)}$ depend only on $\hat{\boldsymbol{A}}^{(2i+3)},\ldots,\hat{\boldsymbol{A}}^{(2M-1)}$ and $\hat{\boldsymbol{B}}^{(2i+2)},\ldots,\hat{\boldsymbol{B}}^{(2M)}$. We therefore have

$$\begin{aligned}\mathbb{E}_{\delta_{s\to t}^{(2i+1)},\zeta_t^{(2i+1)}}\left[\boldsymbol{X}_{A,st}^{(i)} \mid \hat{\boldsymbol{A}}^{(2i+3)},\ldots,\hat{\boldsymbol{A}}^{(2M-1)},\hat{\boldsymbol{B}}^{(2i+2)},\ldots,\hat{\boldsymbol{B}}^{(2M)}\right] = 0,\\ \mathbb{E}_{\delta_{s\to t}^{(2i+1)},\zeta_t^{(2i+1)}}\left[\boldsymbol{Y}_{A,t}^{(i,\xi)} \mid \hat{\boldsymbol{A}}^{(2i+3)},\ldots,\hat{\boldsymbol{A}}^{(2M-1)},\hat{\boldsymbol{B}}^{(2i+2)},\ldots,\hat{\boldsymbol{B}}^{(2M)}\right] = 0,\end{aligned} \tag{28}$$

by the independence of $\delta_{s\to t}^{(2i+1)}$ and $\zeta_t^{(2i+1)}$ in Assumption 4.

C.3 Bound for $\alpha \in (0,1)$ and $\beta \in (0,1)$

We now derive the error bound of VR-HGP for the case when $\alpha,\beta\in(0,1)$. The error bound follows from the next bounds on $\boldsymbol{X}_{B,st}^{(i)}, \boldsymbol{Y}_{B,t}^{(i)}, \boldsymbol{X}_{A,st}^{(i)}$, and $\boldsymbol{Y}_{A,t}^{(i)}$.

Lemma 7. Under Assumption 8, when $\alpha,\beta\in(0,1)$ so that $1-\beta+\beta\eta_A\in(\eta_A,1)$, we have

$$\left\|\boldsymbol{X}_{B,st}^{(i)}\right\|_2^2 \le \frac{\eta_B^2\bar{p}_{ts}^2}{\kappa^2}\left(\frac{1}{2\bar{\delta}_{s\to t}} - 1\right)^2\left(\frac{1-\alpha}{1-\eta_A}\right)^2\alpha^{2(M-i)} + \exp(-O(M)), \tag{29}$$

$$\left\|\boldsymbol{X}_{A,st}^{(i)}\right\|_2^2 \le \frac{\eta_B^2\bar{p}_{ts}^2}{\kappa^2}\left(\frac{1}{2\bar{\delta}_{s\to t}} - 1\right)^2\left(\frac{\eta_A}{1-\eta_A}\frac{(1-\alpha)\beta}{\alpha-(1-\beta+\beta\eta_A)}\right)^2$$
$$\times\left(\alpha^{M-i} - (1-\beta+\beta\eta_A)^{M-i}\right)^2 + \exp(-O(M)), \tag{30}$$

$$\left\|\boldsymbol{Y}_{B,t}^{(i,\xi)}\right\|_2^2 \le \frac{4\eta_B^2}{\kappa^2|\zeta_t^{(2i)}|^2}\left(\frac{1-\alpha}{1-\eta_A}\right)^2\alpha^{2(M-i)} + \exp(-O(M)), \tag{31}$$

$$\left\|\boldsymbol{Y}_{A,t}^{(i,\xi)}\right\|_2^2 \le \frac{4\eta_B^2}{\kappa^2|\zeta_t^{(2i+1)}|^2}\left(\frac{\eta_A}{1-\eta_A}\frac{(1-\alpha)\beta}{\alpha-(1-\beta+\beta\eta_A)}\right)^2\left(\alpha^{M-i} - (1-\beta+\beta\eta_A)^{M-i}\right)^2 + \exp(-O(M)), \tag{32}$$

where

$$\kappa = \max\left\{n, \sum_{s,t} \frac{\bar{p}_{ts}}{\bar{\delta}_{s\to t}}\right\}. \tag{33}$$

Proof.

$$\begin{aligned}
\left\|\boldsymbol{X}_{B,st}^{(i)}\right\|_2^2 &\le \left(\frac{1}{\bar{\delta}_{s\to t}} - 2\right)^2 \bar{p}_{ts}^2 \left\|\hat{\boldsymbol{B}}_t^{\varphi(2i)}\right\|_2^2 \left\|\boldsymbol{R}_3^{(i,M)}\right\|_2^2 \\
&\le \left(\frac{1}{\bar{\delta}_{s\to t}} - 2\right)^2 \bar{p}_{ts}^2 \left(\frac{\eta_B}{2\kappa}\right)^2 \left(\frac{1-\alpha}{1-\eta_A}\alpha^{M-i} + \exp(-O(M))\right)^2 \\
&= \frac{\eta_B^2 \bar{p}_{ts}^2}{\kappa^2} \left(\frac{1}{2\bar{\delta}_{s\to t}} - 1\right)^2 \left(\frac{1-\alpha}{1-\eta_A}\right)^2 \alpha^{2(M-i)} + \exp(-O(M)), \\
\left\|\boldsymbol{X}_{A,st}^{(i)}\right\|_2^2 &\le \left(\frac{1}{\bar{\delta}_{s\to t}} - 2\right)^2 \bar{p}_{ts}^2 \left\|\hat{\boldsymbol{A}}_t^{\varphi(2i+1)}\right\|_2^2 \left(\left\|\hat{\boldsymbol{L}}_4^{(i,M)}\right\|_2 \eta_A^i + \left\|\hat{\boldsymbol{L}}_5^{(i,M)}\right\|_2 \left\|\boldsymbol{R}_1^{(i)}\right\|_2\right)^2 \\
&\le \left(\frac{1}{\bar{\delta}_{s\to t}} - 2\right)^2 \bar{p}_{ts}^2 \left(\frac{\eta_A}{2\kappa}\right)^2 \\
&\quad \times \left(\frac{\eta_B}{1-\eta_A} \frac{(1-\alpha)\beta}{\alpha - (1-\beta+\beta\eta_A)} \left(\alpha^{M-i} - (1-\beta+\beta\eta_A)^{M-i}\right) + \exp(-O(M))\right)^2 \\
&= \frac{\eta_B^2 \bar{p}_{ts}^2}{\kappa^2} \left(\frac{1}{2\bar{\delta}_{s\to t}} - 1\right)^2 \left(\frac{\eta_A}{1-\eta_A} \frac{(1-\alpha)\beta}{\alpha - (1-\beta+\beta\eta_A)}\right)^2 \\
&\quad \times \left(\alpha^{M-i} - (1-\beta+\beta\eta_A)^{M-i}\right)^2 + \exp(-O(M)), \\
\left\|\boldsymbol{Y}_{B,s}^{(i,\xi)}\right\|_2^2 &\le \frac{1}{|\zeta_t^{(2i)}|^2} \sum_{s=1}^n \left\|\mathbb{1}_{ts}\partial_{\boldsymbol{\lambda}_t} g_t(\boldsymbol{y}_t^*, \boldsymbol{\lambda}_t, \xi) + \bar{p}_{ts}\partial_{\boldsymbol{\lambda}_t} h_t(\boldsymbol{y}_t^*, \boldsymbol{\lambda}_t, \xi) - \mathbb{1}_{ts}\boldsymbol{B}_t^{\psi} - \bar{p}_{ts}\boldsymbol{B}_t^{\varphi}\right\|_2^2 \left\|\boldsymbol{R}_3^{(i,M)}\right\|_2^2 \\
&\le \frac{1}{|\zeta_s^{(2t)}|^2} \left(\frac{\eta_B}{2\kappa}\right)^2 \left(2 + 2\underbrace{\sum_{s=1}^n \bar{p}_{ts}}_{=1}\right)^2 \left(\frac{1-\alpha}{1-\eta_A}\alpha^{M-i} + \exp(-O(M))\right)^2 \\
&= \frac{4\eta_B^2}{\kappa^2 |\zeta_t^{(2i)}|^2} \left(\frac{1-\alpha}{1-\eta_A}\right)^2 \alpha^{2(M-i)} + \exp(-O(M)), \\
\left\|\boldsymbol{Y}_{A,s}^{(i,\xi)}\right\|_2^2 &\le \frac{1}{|\zeta_s^{(2i+1)}|^2} \sum_{s=1}^n \left\|\mathbb{1}_{ts}\partial_{\boldsymbol{y}_t} g_t(\boldsymbol{y}_t^*, \boldsymbol{\lambda}_t, \xi) + \bar{p}_{ts}\partial_{\boldsymbol{y}_t} h_t(\boldsymbol{y}_t^*, \boldsymbol{\lambda}_t, \xi) - \mathbb{1}_{ts}\boldsymbol{A}_t^{\psi} - \bar{p}_{ts}\boldsymbol{A}_t^{\varphi}\right\|_2^2 \\
&\quad \times \left(\left\|\hat{\boldsymbol{L}}_4^{(i,M)}\right\|_2 \eta_{\boldsymbol{A}}^i + \left\|\hat{\boldsymbol{L}}_5^{(i,M)}\right\|_2 \left\|\boldsymbol{R}_1^{(i)}\right\|_2\right)^2 \\
&\le \frac{1}{|\zeta_t^{(2i+1)}|^2} \left(\frac{\eta_A}{2\kappa}\right)^2 \left(2 + 2\underbrace{\sum_{s=1}^n \bar{p}_{ts}}_{=1}\right)^2 \\
&\quad \times \left(\frac{\eta_B}{1-\eta_A} \frac{(1-\alpha)\beta}{\alpha - (1-\beta+\beta\eta_A)} \left(\alpha^{M-i} - (1-\beta+\beta\eta_A)^{M-i}\right) + \exp(-O(M))\right)^2 \\
&= \frac{4\eta_B^2}{\kappa^2 |\zeta_t^{(2i+1)}|^2} \left(\frac{\eta_A}{1-\eta_A} \frac{(1-\alpha)\beta}{\alpha - (1-\beta+\beta\eta_A)}\right)^2 \left(\alpha^{M-i} - (1-\beta+\beta\eta_A)^{M-i}\right)^2 + \exp(-O(M)).
\end{aligned}$$

□

Theorem 8. Suppose Assumptions 1–8 hold true, and $|\zeta_t^{(2i)}| = |\zeta_t^{(2i+1)}| = b$ for any t and i. Then, with probability at least $1-\epsilon$, we have

$$\begin{aligned}&\left\|\boldsymbol{v}^{(M+1)} + \boldsymbol{c}^{\boldsymbol{\lambda}} - \mathrm{d}_{\boldsymbol{\lambda}_j} F(\boldsymbol{y}^*, \boldsymbol{\lambda})\right\| \\ &\leq \mu_{\alpha,\beta}\tau \sqrt{\left(\sum_{s,t=1}^{n} \bar{p}_{ts}^2 \left(\frac{1}{2\bar{\delta}_{s\to t}} - 1\right)^2 + \frac{4n}{b}\right) \log \frac{n(d_{\boldsymbol{y}} + d_{\boldsymbol{\lambda}})}{\epsilon}} + \exp(-O(M)),\end{aligned}$$

where

$$\mu_{\alpha,\beta} = \sqrt{8\frac{1-\alpha}{1+\alpha}\left(1 + \frac{1+\alpha(1-\beta+\beta\eta_A)}{1-\alpha(1-\beta+\beta\eta_A)}\frac{\beta^2\eta_A^2}{1-(1-\beta+\beta\eta_A)^2}\right)}, \quad \tau = \frac{\eta_B\|\boldsymbol{c}^{\boldsymbol{y}}\|}{\kappa(1-\eta_A)}.$$

Proof. We first have

$$\left\|\boldsymbol{v}^{(M+1)} + \boldsymbol{c}^{\boldsymbol{\lambda}} - \mathrm{d}_{\boldsymbol{\lambda}_j} F(\boldsymbol{y}^*, \boldsymbol{\lambda})\right\| \leq \left\|\boldsymbol{v}^{(M+1)} - \boldsymbol{B}\sum_{i=0}^{M} \boldsymbol{A}^i \boldsymbol{c}^{\boldsymbol{y}}\right\| + \left\|\boldsymbol{B}\sum_{i=M+1}^{\infty} \boldsymbol{A}^i \boldsymbol{c}^{\boldsymbol{y}}\right\|.$$

Here, we can bound the second term by

$$\left\|\boldsymbol{B}\sum_{i=M+1}^{\infty} \boldsymbol{A}^i \boldsymbol{c}^{\boldsymbol{y}}\right\| \leq \eta_B\|\boldsymbol{c}^{\boldsymbol{y}}\| \sum_{i=M+1}^{\infty} \eta_A^i \leq \frac{\eta_B\|\boldsymbol{c}^{\boldsymbol{y}}\|}{1-\eta_A}\eta_A^{M+1} = \exp\left(-O(M)\right).$$

The conditions (28) ensure that we can bound the first term by using Matrix Azuma's inequality; with probability at least $1-\epsilon$, we have

$$\left\|\boldsymbol{v}^{(M+1)} - \boldsymbol{B}\sum_{i=0}^{M} \boldsymbol{A}^i \boldsymbol{c}^{\boldsymbol{y}}\right\| \leq \sqrt{8\sigma^2 \frac{n(d_{\boldsymbol{y}} + d_{\boldsymbol{\lambda}})}{\epsilon}},$$

where

$$\begin{aligned}\frac{\sigma^2}{\|\boldsymbol{c}^{\boldsymbol{y}}\|^2} &\leq \sum_{i=0}^{M}\sum_{s,t=1}^{n} \left\|\boldsymbol{X}_{B,st}^{(i)}\right\|_2^2 + \sum_{i=0}^{M-1}\sum_{s,t=1}^{n} \left\|\boldsymbol{X}_{A,st}^{(i)}\right\|_2^2 \\ &\quad + \sum_{i=0}^{M}\sum_{t=1}^{n}\sum_{\xi\in\zeta_t^{(2i)}} \left\|\boldsymbol{Y}_{B,t}^{(i,\xi)}\right\|_2^2 + \sum_{i=0}^{M-1}\sum_{t=1}^{n}\sum_{\xi\in\zeta_t^{(2i+1)}} \left\|\boldsymbol{Y}_{A,t}^{(i,\xi)}\right\|_2^2 \\ &\leq \sum_{i=0}^{M}\sum_{s,t=1}^{n} \frac{\eta_B^2\bar{p}_{ts}^2}{\kappa^2}\left(\frac{1}{2\bar{\delta}_{s\to t}} - 1\right)^2 \left(\frac{1-\alpha}{1-\eta_A}\right)^2 \alpha^{2(M-i)} \\ &\quad + \sum_{i=0}^{M-1}\sum_{s,t=1}^{n} \frac{\eta_B^2\bar{p}_{ts}^2}{\kappa^2}\left(\frac{1}{2\bar{\delta}_{s\to t}} - 1\right)^2 \left(\frac{\eta_A}{1-\eta_A}\frac{(1-\alpha)\beta}{\alpha-(1-\beta+\beta\eta_A)}\right)^2 \left(\alpha^{M-i} - (1-\beta+\beta\eta_A)^{M-i}\right)^2 \\ &\quad + \sum_{i=0}^{M}\sum_{t=1}^{n}\sum_{\xi\in\zeta_t^{(2i)}} \frac{4\eta_B^2}{\kappa^2|\zeta_t^{(2i)}|^2}\left(\frac{1-\alpha}{1-\eta_A}\right)^2 \alpha^{2(M-i)} \\ &\quad + \sum_{i=0}^{M-1}\sum_{t=1}^{n}\sum_{\xi\in\zeta_t^{(2i+1)}} \frac{4\eta_B^2}{\kappa^2|\zeta_t^{(2i+1)}|^2}\left(\frac{\eta_A}{1-\eta_A}\frac{(1-\alpha)\beta}{\alpha-(1-\beta+\beta\eta_A)}\right)^2 \left(\alpha^{M-i} - (1-\beta+\beta\eta_A)^{M-i}\right)^2 \\ &\quad + \exp(-O(M))\end{aligned}$$

$$
\begin{aligned}
&= \frac{\eta_B^2}{\kappa^2}\left[\sum_{s,t=1}^{n} \bar{p}_{ts}^2 \left(\frac{1}{2\bar{\delta}_{s\to t}}-1\right)^2\right]\left(\frac{1-\alpha}{1-\eta_A}\right)^2 \frac{1-\alpha^{2(M+1)}}{1-\alpha^2} \\
&\quad + \frac{\eta_B^2}{\kappa^2}\left[\sum_{s,t=1}^{n} \bar{p}_{ts}^2 \left(\frac{1}{2\bar{\delta}_{s\to t}}-1\right)^2\right]\left(\frac{\eta_A}{1-\eta_A}\frac{(1-\alpha)\beta}{\alpha-(1-\beta+\beta\eta_A)}\right)^2 \\
&\qquad \times \left(\frac{\alpha^2-\alpha^{2(M+1)}}{1-\alpha^2} + \frac{(1-\beta+\beta\eta_A)^2-(1-\beta+\beta\eta_A)^{2(M+1)}}{1-(1-\beta+\beta\eta_A)^2} - 2\frac{\alpha(1-\beta+\beta\eta_A)-\alpha^{M+1}(1-\beta+\beta\eta_A)^{M+1}}{1-\alpha(1-\beta+\beta\eta_A)}\right) \\
&\quad + \frac{\eta_B^2}{\kappa^2}\left[\sum_{t=1}^{n} \frac{4}{|\zeta_t^{(2i)}|}\right]\left(\frac{1-\alpha}{1-\eta_A}\right)^2 \frac{1-\alpha^{2(M+1)}}{1-\alpha^2} \\
&\quad + \frac{\eta_B^2}{\kappa^2}\left[\sum_{s=1}^{K} \frac{4}{|\zeta_t^{(2i+1)}|}\right]\left(\frac{\eta_A}{1-\eta_A}\frac{(1-\alpha)\beta}{\alpha-(1-\beta+\beta\eta_A)}\right)^2 \\
&\qquad \times \left(\frac{\alpha^2-\alpha^{2(M+1)}}{1-\alpha^2} + \frac{(1-\beta+\beta\eta_A)^2-(1-\beta+\beta\eta_A)^{2(M+1)}}{1-(1-\beta+\beta\eta_A)^2} - 2\frac{\alpha(1-\beta+\beta\eta_A)-\alpha^{M+1}(1-\beta+\beta\eta_A)^{M+1}}{1-\alpha(1-\beta+\beta\eta_A)}\right) \\
&\quad + \exp(-O(M)) \\
&\le \frac{\eta_B^2}{\kappa^2}\left[\sum_{s,t=1}^{n} \bar{p}_{ts}^2 \left(\frac{1}{2\bar{\delta}_{s\to t}}-1\right)^2 + \sum_{t=1}^{n}\frac{4}{|\zeta_t^{(2i)}|}\right]\left(\frac{1-\alpha}{1-\eta_A}\right)^2 \frac{1}{1-\alpha^2} \\
&\quad + \frac{\eta_B^2}{\kappa^2}\left[\sum_{s,t=1}^{n} \bar{p}_{ts}^2 \left(\frac{1}{2\bar{\delta}_{s\to t}}-1\right)^2 + \sum_{t=1}^{n}\frac{4}{|\zeta_t^{(2i+1)}|}\right]\left(\frac{\eta_A}{1-\eta_A}\frac{(1-\alpha)\beta}{\alpha-(1-\beta+\beta\eta_A)}\right)^2 \\
&\qquad \times \left(\frac{\alpha^2}{1-\alpha^2} + \frac{(1-\beta+\beta\eta_A)^2}{1-(1-\beta+\beta\eta_A)^2} - 2\frac{\alpha(1-\beta+\beta\eta_A)}{1-\alpha(1-\beta+\beta\eta_A)}\right) \\
&\quad + \exp(-O(M)) \\
&= \frac{\eta_B^2}{\kappa^2}\left[\sum_{s,t=1}^{n} \bar{p}_{ts}^2 \left(\frac{1}{2\bar{\delta}_{s\to t}}-1\right)^2 + \sum_{t=1}^{n}\frac{4}{|\zeta_t^{(2i)}|}\right]\left(\frac{1-\alpha}{1-\eta_A}\right)^2 \frac{1}{1-\alpha^2} \\
&\quad + \frac{\eta_B^2}{\kappa^2}\left[\sum_{s,t=1}^{n} \bar{p}_{ts}^2 \left(\frac{1}{2\bar{\delta}_{s\to t}}-1\right)^2 + \sum_{t=1}^{n}\frac{4}{|\zeta_t^{(2i+1)}|}\right]\left(\frac{\eta_A}{1-\eta_A}\frac{(1-\alpha)\beta}{\alpha-(1-\beta+\beta\eta_A)}\right)^2 \\
&\qquad \times \frac{(1+\alpha(1-\beta+\beta\eta_A))(\alpha-(1-\beta+\beta\eta_A))^2}{(1-\alpha^2)(1-(1-\beta+\beta\eta_A)^2)(1-\alpha(1-\beta+\beta\eta_A))} \\
&\quad + \exp(-O(M)) \\
&= \frac{\eta_B^2}{\kappa^2}\left[\sum_{s,t=1}^{n} \bar{p}_{ts}^2 \left(\frac{1}{2\bar{\delta}_{s\to t}}-1\right)^2 + \sum_{t=1}^{n}\frac{4}{|\zeta_t^{(2i)}|}\right]\left(\frac{1}{1-\eta_A}\right)^2 \frac{1-\alpha}{1+\alpha} \\
&\quad + \frac{\eta_B^2}{\kappa^2}\left[\sum_{s,t=1}^{n} \bar{p}_{ts}^2 \left(\frac{1}{2\bar{\delta}_{s\to t}}-1\right)^2 + \sum_{t=1}^{n}\frac{4}{|\zeta_t^{(2i+1)}|}\right]\left(\frac{1}{1-\eta_A}\right)^2 \frac{1-\alpha}{1+\alpha} \\
&\qquad \times \frac{1+\alpha(1-\beta+\beta\eta_A)}{1-\alpha(1-\beta+\beta\eta_A)}\frac{\beta^2\eta_A^2}{1-(1-\beta+\beta\eta_A)^2} \\
&\quad + \exp(-O(M)).
\end{aligned}
$$

When $|\zeta_t^{(2i)}| = |\zeta_t^{(2i+1)}| = b$ for any t and i, we further have

$$
\begin{aligned}
\sigma^2 &\le \frac{\eta_B^2\|\boldsymbol{c}^{\boldsymbol{y}}\|^2}{\kappa^2(1-\eta_A)^2}\left[\sum_{s,t=1}^{n} \bar{p}_{ts}^2 \left(\frac{1}{2\bar{\delta}_{s\to t}}-1\right)^2 + \frac{4n}{b}\right] \\
&\qquad \times \frac{1-\alpha}{1+\alpha}\left(1+\frac{1+\alpha(1-\beta+\beta\eta_A)}{1-\alpha(1-\beta+\beta\eta_A)}\frac{\beta^2\eta_A^2}{1-(1-\beta+\beta\eta_A)^2}\right) + \exp(-O(M)).
\end{aligned}
$$

□

C.4 BOUND FOR $\alpha = 1$ AND $\beta = 0$

Setting $\alpha = 1$ and $\beta = 0$ recovers naive HGP. Here, we derive the error bound for naive HGP.

Lemma 9. Under Assumption 8, when $\alpha = 1$ and $\beta = 0$ so that $1 - \beta + \beta\eta_A = 1$, we have

$$\left\|\boldsymbol{X}_{B,st}^{(i)}\right\|_2^2 \leq \frac{\eta_B^2 \bar{p}_{ts}^2}{\kappa^2}\left(\frac{1}{2\bar{\delta}_{s\to t}} - 1\right)^2 \eta_A^{2i}, \tag{34}$$

$$\left\|\boldsymbol{X}_{A,st}^{(i)}\right\|_2^2 \leq \frac{\eta_B^2 \bar{p}_{ts}^2}{\kappa^2}\left(\frac{1}{2\bar{\delta}_{s\to t}} - 1\right)^2 \left(\frac{\eta_A}{1-\eta_A}\right)^2 \eta_A^{2i} + \exp\left(-O(M)\right), \tag{35}$$

$$\left\|\boldsymbol{Y}_{B,t}^{(i,\xi)}\right\|_2^2 \leq \frac{4\eta_B^2}{\kappa^2 |\zeta_t^{(2i)}|^2}\eta_A^{2i}, \tag{36}$$

$$\left\|\boldsymbol{Y}_{A,t}^{(i,\xi)}\right\|_2^2 \leq \frac{4\eta_B^2}{\kappa^2 |\zeta_t^{(2i+1)}|^2}\left(\frac{\eta_A}{1-\eta_A}\right)^2 \eta_A^{2i} + \exp\left(-O(M)\right). \tag{37}$$

Proof.

$$\begin{aligned}
\left\|\boldsymbol{X}_{B,st}^{(i)}\right\|_2^2 &\leq \left(\frac{1}{\bar{\delta}_{s\to t}} - 2\right)^2 \bar{p}_{ts}^2 \left\|\hat{\boldsymbol{B}}_t^{\varphi(2i)}\right\|_2^2 \left\|\boldsymbol{R}_3^{(i,M)}\right\|_2^2 \\
&\leq \left(\frac{1}{\bar{\delta}_{s\to t}} - 2\right)^2 \bar{p}_{ts}^2 \left(\frac{\eta_B}{2\kappa}\right)^2 \left(\eta_A^i\right)^2 \\
&= \frac{\eta_B^2 \bar{p}_{ts}^2}{\kappa^2}\left(\frac{1}{2\bar{\delta}_{s\to t}} - 1\right)^2 \eta_A^{2i}, \\
\left\|\boldsymbol{X}_{A,st}^{(i)}\right\|_2^2 &\leq \left(\frac{1}{\bar{\delta}_{s\to t}} - 2\right)^2 \bar{p}_{ts}^2 \left\|\hat{\boldsymbol{A}}_t^{\varphi(2i+1)}\right\|_2^2 \left(\left\|\hat{\boldsymbol{L}}_4^{(i,M)}\right\|_2 \eta_A^i + \left\|\hat{\boldsymbol{L}}_5^{(i,M)}\right\|_2 \left\|\boldsymbol{R}_1^{(i)}\right\|_2\right)^2 \\
&\leq \left(\frac{1}{\bar{\delta}_{s\to t}} - 2\right)^2 \bar{p}_{ts}^2 \left(\frac{\eta_A}{2\kappa}\right)^2 \left(\frac{\eta_B}{1-\eta_A}(\eta_A^i - \eta_A^M)\right)^2 \\
&= \frac{\eta_B^2 \bar{p}_{ts}^2}{\kappa^2}\left(\frac{1}{2\bar{\delta}_{s\to t}} - 1\right)^2 \left(\frac{\eta_A}{1-\eta_A}\right)^2 \eta_A^{2i} + \exp\left(-O(M)\right), \\
\left\|\boldsymbol{Y}_{B,s}^{(i,\xi)}\right\|_2^2 &\leq \frac{1}{|\zeta_t^{(2i)}|^2}\sum_{s=1}^n \left\|\mathbb{1}_{ts}\partial_{\boldsymbol{\lambda}_t} g_t(\boldsymbol{y}_t^*, \boldsymbol{\lambda}_t, \xi) + \bar{p}_{ts}\partial_{\boldsymbol{\lambda}_t} h_t(\boldsymbol{y}_t^*, \boldsymbol{\lambda}_t, \xi) - \mathbb{1}_{ts}\boldsymbol{B}_t^{\psi} - \bar{p}_{ts}\boldsymbol{B}_t^{\varphi}\right\|_2^2 \left\|\boldsymbol{R}_3^{(i,M)}\right\|_2^2 \\
&\leq \frac{1}{|\zeta_s^{(2t)}|^2}\left(\frac{\eta_B}{2\kappa}\right)^2 \left(2 + 2\sum_{s=1}^n \bar{p}_{ts}\right)^2 \left(\eta_A^i\right)^2 \\
&= \frac{4\eta_B^2}{\kappa^2 |\zeta_t^{(2i)}|^2}\eta_A^{2i}, \\
\left\|\boldsymbol{Y}_{A,s}^{(i,\xi)}\right\|_2^2 &\leq \frac{1}{|\zeta_s^{(2i+1)}|^2}\sum_{s=1}^n \left\|\mathbb{1}_{ts}\partial_{\boldsymbol{y}_t} g_t(\boldsymbol{y}_t^*, \boldsymbol{\lambda}_t, \xi) + \bar{p}_{ts}\partial_{\boldsymbol{y}_t} h_t(\boldsymbol{y}_t^*, \boldsymbol{\lambda}_t, \xi) - \mathbb{1}_{ts}\boldsymbol{A}_t^{\psi} - \bar{p}_{ts}\boldsymbol{A}_t^{\varphi}\right\|_2^2 \\
&\quad \times \left(\left\|\hat{\boldsymbol{L}}_4^{(i,M)}\right\|_2 \eta_{\boldsymbol{A}}^i + \left\|\hat{\boldsymbol{L}}_5^{(i,M)}\right\|_2 \left\|\boldsymbol{R}_1^{(i)}\right\|_2\right)^2 \\
&\leq \frac{1}{|\zeta_t^{(2i+1)}|^2}\left(\frac{\eta_A}{2\kappa}\right)^2 \left(2 + 2\sum_{s=1}^n \bar{p}_{ts}\right)^2 \left(\frac{\eta_B}{1-\eta_A}(\eta_A^i - \eta_A^M)\right)^2 \\
&= \frac{4\eta_B^2}{\kappa^2 |\zeta_t^{(2i+1)}|^2}\left(\frac{\eta_A}{1-\eta_A}\right)^2 \eta_A^{2i} + \exp\left(-O(M)\right).
\end{aligned}$$

□

Theorem 10. Suppose Assumptions 1–8 hold true, and $|\zeta_t^{(2i)}| = |\zeta_t^{(2i+1)}| = b$ for any t and i. When $\alpha = 1, \beta = 0$, with probability at least $1-\epsilon$, we have

$$\left\|\boldsymbol{v}^{(M+1)} - \mathrm{d}_{\boldsymbol{\lambda}_j} F(\boldsymbol{y}^*, \boldsymbol{\lambda})\right\| \leq \mu_{1,0}\tau \sqrt{\left(\sum_{s,t=1}^{n} \bar{p}_{ts}^2 \left(\frac{1}{2\bar{\delta}_{s\to t}} - 1\right)^2 + \frac{4n}{|\zeta|}\right) \log \frac{n(d_{\boldsymbol{y}} + d_{\boldsymbol{\lambda}})}{\epsilon}} + \exp(-O(M)),$$

where

$$\mu_{1,0} = \sqrt{8\frac{\eta_A^2 + (1-\eta_A)^2}{1-\eta_A^2}}, \quad \tau = \frac{\eta_B \|\boldsymbol{c}^{\boldsymbol{y}}\|}{\kappa(1-\eta_A)}.$$

Proof. We first have

$$\left\|\boldsymbol{v}^{(M+1)} - \mathrm{d}_{\boldsymbol{\lambda}_j} F(\boldsymbol{y}^*, \boldsymbol{\lambda})\right\| \leq \left\|\boldsymbol{v}^{(M+1)} - \boldsymbol{B}\sum_{i=0}^{M} \boldsymbol{A}^i \boldsymbol{c}^{\boldsymbol{y}}\right\| + \left\|\boldsymbol{B}\sum_{i=M+1}^{\infty} \boldsymbol{A}^i \boldsymbol{c}^{\boldsymbol{y}}\right\|.$$

Here, we can bound the second term by

$$\left\|\boldsymbol{B}\sum_{i=M+1}^{\infty} \boldsymbol{A}^i \boldsymbol{c}^{\boldsymbol{y}}\right\| \leq \eta_B \|\boldsymbol{c}^{\boldsymbol{y}}\| \sum_{i=M+1}^{\infty} \eta_A^i \leq \frac{\eta_B \|\boldsymbol{c}^{\boldsymbol{y}}\|}{1-\eta_A} \eta_A^{M+1} = \exp\left(-O(M)\right).$$

We can bound the first term by using Matrix Azuma's inequality; with probability at least $1-\epsilon$, we have

$$\left\|\boldsymbol{v}^{(M+1)} - \boldsymbol{B}\sum_{i=0}^{M} \boldsymbol{A}^i \boldsymbol{c}^{\boldsymbol{y}}\right\| \leq \sqrt{8\sigma^2 \frac{n(d_{\boldsymbol{y}} + d_{\boldsymbol{\lambda}})}{\epsilon}},$$

where

$$\begin{aligned}
\frac{\sigma^2}{\|\boldsymbol{c}^{\boldsymbol{y}}\|^2} &\leq \sum_{i=0}^{M}\sum_{s,t=1}^{n} \left\|\boldsymbol{X}_{B,st}^{(i)}\right\|_2^2 + \sum_{i=0}^{M-1}\sum_{s,t=1}^{n} \left\|\boldsymbol{X}_{A,st}^{(i)}\right\|_2^2 \\
&\quad + \sum_{i=0}^{M}\sum_{t=1}^{n}\sum_{\xi\in\zeta_t^{(2i)}} \left\|\boldsymbol{Y}_{B,t}^{(i,\xi)}\right\|_2^2 + \sum_{i=0}^{M-1}\sum_{t=1}^{n}\sum_{\xi\in\zeta_t^{(2i+1)}} \left\|\boldsymbol{Y}_{A,t}^{(i,\xi)}\right\|_2^2 \\
&\leq \sum_{i=0}^{M}\sum_{s,t=1}^{n} \frac{\eta_B^2 \bar{p}_{ts}^2}{\kappa^2}\left(\frac{1}{2\bar{\delta}_{s\to t}} - 1\right)^2 \eta_A^{2i} + \sum_{i=0}^{M-1}\sum_{s,t=1}^{n} \frac{\eta_B^2 \bar{p}_{ts}^2}{\kappa^2}\left(\frac{1}{2\bar{\delta}_{s\to t}} - 1\right)^2 \left(\frac{\eta_A}{1-\eta_A}\right)^2 \eta_A^{2i} \\
&\quad + \sum_{i=0}^{M}\sum_{t=1}^{n}\sum_{\xi\in\zeta_t^{(2i)}} \frac{4\eta_B^2}{\kappa^2|\zeta_t^{(2i)}|^2}\eta_A^{2i} + \sum_{i=0}^{M-1}\sum_{t=1}^{n}\sum_{\xi\in\zeta_t^{(2i+1)}} \frac{4\eta_B^2}{\kappa^2|\zeta_t^{(2i+1)}|^2}\left(\frac{\eta_A}{1-\eta_A}\right)^2 \eta_A^{2i} + \exp(-O(M)) \\
&= \frac{\eta_B^2}{\kappa^2}\left[\sum_{s,t=1}^{n} \bar{p}_{ts}^2 \left(\frac{1}{2\bar{\delta}_{s\to t}} - 1\right)^2 + \sum_{t=1}^{n} \frac{4}{|\zeta_t^{(2i)}|}\right] \frac{1-\eta_A^{2(M+1)}}{1-\eta_A^2} \\
&\quad + \frac{\eta_B^2}{\kappa^2}\left[\sum_{s,t=1}^{n} \bar{p}_{ts}^2 \left(\frac{1}{2\bar{\delta}_{s\to t}} - 1\right)^2 + \sum_{t=1}^{n} \frac{4}{|\zeta_t^{(2i+1)}|}\right] \left(\frac{\eta_A}{1-\eta_A}\right)^2 \frac{1-\eta_A^{2M}}{1-\eta_A^2} + \exp(-O(M)) \\
&= \frac{\eta_B^2}{\kappa^2}\left[\sum_{s,t=1}^{n} \bar{p}_{ts}^2 \left(\frac{1}{2\bar{\delta}_{s\to t}} - 1\right)^2 + \sum_{t=1}^{n} \frac{4}{|\zeta_t^{(2i)}|}\right] \frac{1}{1-\eta_A^2} \\
&\quad + \frac{\eta_B^2}{\kappa^2}\left[\sum_{s,t=1}^{n} \bar{p}_{ts}^2 \left(\frac{1}{2\bar{\delta}_{s\to t}} - 1\right)^2 + \sum_{t=1}^{n} \frac{4}{|\zeta_t^{(2i+1)}|}\right] \left(\frac{\eta_A}{1-\eta_A}\right)^2 \frac{1}{1-\eta_A^2} + \exp(-O(M)).
\end{aligned}$$

When $|\zeta_t^{(2i)}| = |\zeta_t^{(2i+1)}| = b$ for any t and i, we further have

$$\sigma^2 \leq \frac{\eta_B^2 \|\boldsymbol{c}^{\boldsymbol{y}}\|^2}{\kappa^2(1-\eta_A^2)}\left(1 + \frac{\eta_A^2}{(1-\eta_A)^2}\right)\left[\sum_{s,t=1}^{n} \bar{p}_{ts}^2 \left(\frac{1}{2\bar{\delta}_{s\to t}} - 1\right)^2 + \frac{4n}{b}\right] + \exp(-O(M)).$$

□

C.5 COMPARISON OF $\mu_{\alpha,\beta}$ AND $\mu_{0,1}$

The estimation errors of VR-HGP and naive HGP are dominated by their scaling factors.

$$\mu_{\alpha,\beta} = \sqrt{8\frac{1-\alpha}{1+\alpha}\left(1+\frac{1+\alpha(1-\beta+\beta\eta_A)}{1-\alpha(1-\beta+\beta\eta_A)}\frac{\beta^2\eta_A^2}{1-(1-\beta+\beta\eta_A)^2}\right)},$$

$$\mu_{1,0} = \sqrt{8\frac{\eta_A^2+(1-\eta_A)^2}{1-\eta_A^2}}.$$

Figure 1 shows that $\mu_{\alpha,\beta}$ is a few times smaller than $\mu_{1,0}$ for any $\eta_A \in (0,1)$ if we choose α close to one and β close to zero. This result indicates that the error of VR-HGP can be a few times smaller than the one of naive HGP for sufficiently large M where the diminishing term $\exp(-O(M))$ is negligibly small.

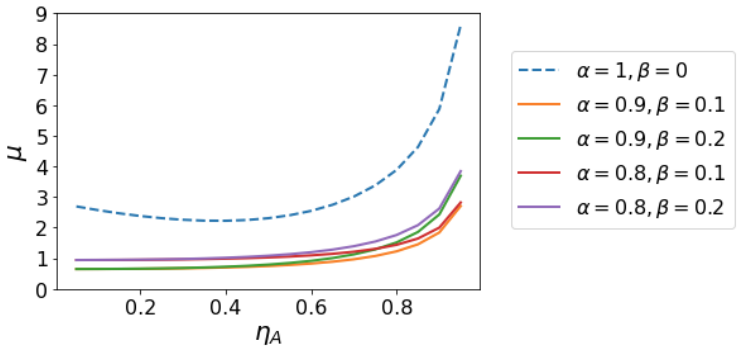

Figure 1: Comparisons of $\mu_{\alpha,\beta}$ and $\mu_{1,0}$ for $\eta_A \in (0,1)$.

C.6 COMPARISON OF α AND β

We empirically evaluated the advantages of VR-HGP in stochastic communications as well as found that $(\alpha,\beta) = (0.9, 0.1)$ performed well in practice.

We compared the ℓ_2 norm between of the hyper-gradient estimation $\boldsymbol{v}^{(m)}$ at the m-th round of HGP and the true hyper-gradient $\mathrm{d}_{\boldsymbol{\lambda}}\bar{F}(\boldsymbol{x}^*, \boldsymbol{\lambda})$ which computed using the explicit $(\boldsymbol{I}-\bar{\boldsymbol{A}})^{-1}$. We made a synthetic one-dimensional dataset with two classes by randomly selecting two digits from MNIST and averaging the inputs of each sample. We let $n = 3$ clients performed 500 iterations of Eq. (4b) ensuring the convergence of SGP. For all $i \in [n]$, we used the binary cross-entropy loss for f_i and F_i computed on local training and validation datasets with 100 samples, respectively. We adopted `StoU` communication network presented in Section 6. In order to purely evaluate the effect of edge stochasticity $\delta_{j\to i}^{(m)}/\bar{\delta}_{j\to i}$, which we pointed the source of the high variance in Section 4.2, we excluded the randomness of minibatches ζ by adopting $|\zeta_i^{(t)}| = 100$ for all time steps in SGP and HGP and by using the true $\bar{p}_{ij}$ and $\bar{\delta}_{j\to i}$ for all $i, j \in [n]$. We computed $\mathrm{d}_{\boldsymbol{\lambda}}\bar{F}(\boldsymbol{x}^*, \boldsymbol{\lambda})$ from the explicit computation of $\bar{\boldsymbol{B}}\left(\boldsymbol{I}-\bar{\boldsymbol{A}}\right)^{-1}\boldsymbol{c}^{\boldsymbol{y}} + \boldsymbol{c}^{\boldsymbol{\lambda}}$ using expected values of $\bar{p}_{ij}$ and $\bar{\delta}_{j\to i}$ for all $i, j \in [n]$. The HGP was conducted to obtain $\boldsymbol{v}^{(m)}$ after the iterations of SGP using $M = 500$ and the alternative samplings, i.e., $\tilde{\boldsymbol{A}}^{(2m+1)}$ and $\tilde{\boldsymbol{B}}^{(2m)}$ for $m = 0, \ldots, M-1$.

Fig. 2 shows VR-HGP with $(\alpha,\beta) = (0.9, 0.1)$ provided the smallest estimation error and the larger number of estimation rounds tends to have smaller error. However HGP, which is a special case of VR-HGP with, failed to attain smaller error than the well-tuned VR-HGP with $(\alpha,\beta) = (0.9, 0.1)$, as shown in the case where $\alpha = 1.0, \beta = 0.0$ in Fig. 2. This larger estimation error was also observed in experiments with different random seeds. We also observed that HGP could not reduce the estimation error after around $m = 5$ indicating the larger number of rounds does not always help the better estimation in HGP on stochastic communication networks.

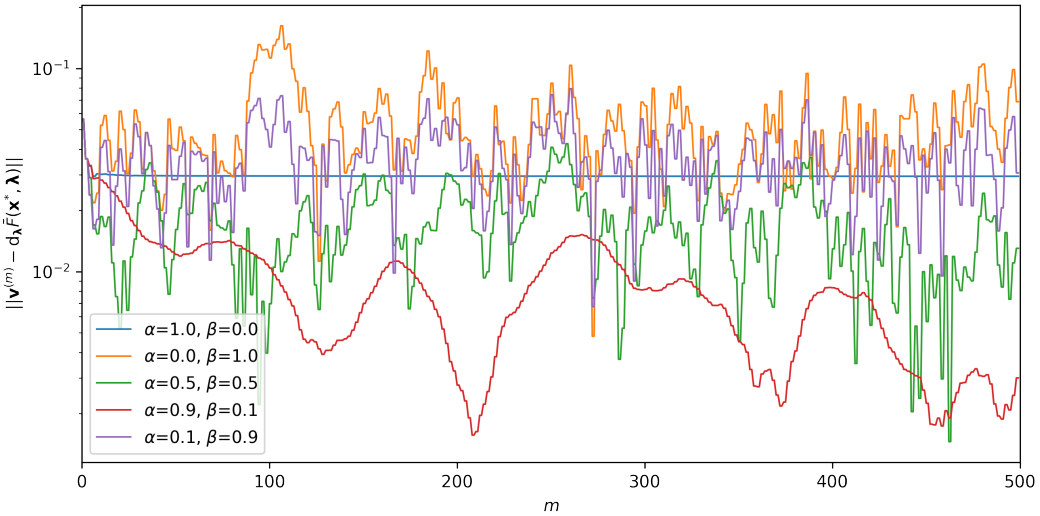

Figure 2: ℓ_2 norm between the estimation of VR-HGP $\boldsymbol{v}^{(m)}$ and the true hyper-gradient $\mathrm{d}_{\boldsymbol{\lambda}}\bar{F}\left(\boldsymbol{x}^*, \boldsymbol{\lambda}\right)$ at the m-th estimation round with different combinations of α and β.

D Detailed Experimental Settings

The experiments in Section 6 followed the settings of EMNIST (Cohen et al., 2017) classification in Marfoq et al. (2021), unless otherwise mentioned.

Communication networks We simulated four communication networks on which the clients perform the distributed learning: fully-connected (`FC`), static undirected (`FixU`), stochastic undirected (`StoU`), and stochastic directed (`StoD`).

`FC` allows clients to communicate with all the other clients in all the time steps, i.e. $\delta_{i \to j}^{(t)} = 1$ for all $i, j \in [n]$ and $t \in \mathbb{N}$. `FixU` uses time-invariant and sparse undirected communication network simulated by a binomial Erdős-Rényi graph (Erdős & Rényi, 1959) with parameter $p = 0.4$ adding the self-loop edges. Following the setting in Marfoq et al. (2021), we generated a doubly stochastic mixing matrix using the Fast Mixing Markov Chain (Boyd et al., 2003) rule. `StoU` uses stochastic and undirected network in which any undirected edge $\delta_{j \to i}^{(t)} = \delta_{i \to j}^{(t)}$ independently realizes at each step with the probability $\bar{\delta}_{j \to i} \in [0, 1]$. In `StoD` each direction of edges $\delta_{j \to i}^{(t)}$ are independently sampled at probability $\bar{\delta}_{j \to i}$, forming stochastic and directed network. `StoD` forms the asymmetric expected mixing matrix given by the `StoD` network is asymmetric representing the communication bias between the clients; some clients may communicate more infrequently than others due to bottlenecks in physical network environments or long computation times of local updates due to poor computational resources. We sampled $\bar{\delta}_{j \to i}$ from the uniform distribution with $[0.4, 0.8]$ both in `StoU` and `StoD`

Proposed approaches We solved personalization of classification models using three different formulation: PDBO-MTL, PDBO-DA, and PDBO-MTL&DA.

For PDBO-DA, we optimize the pseudo sampling rate to recover data augmentation-based personalization (Duan et al., 2019; Zhao et al., 2018). PDBO-DA optimize $\boldsymbol{\lambda}_i^C \in \mathbb{R}^C$ to learn the label-wise weight vector $C\texttt{Softmax}\left(\boldsymbol{\lambda}_i\right) \in [0, C]^C$. In the inner-problem, the losses of instances labeled as $c \in [C]$ are multiplied by the c-th element of the weight vector.

PDBO-MTL is obtained by applying PDBO to FedEM Marfoq et al. (2021). PDBO-MTL lets each client train an ensemble classifier that outputs weighted average predictions across $K = 3$ of CNNs. We trained CNN parameters as the inner-problem and optimized the hyperparameters $\boldsymbol{\lambda}_i^K \in \mathbb{R}^K$ to obtain ensemble weight vector $\texttt{Softmax}\left(\boldsymbol{\lambda}_i\right) \in [0, 1]^K$.

PDBO-MTL&DA combines PDBO-DA and PDBO-MTL optimizing $[\boldsymbol{\lambda}_i^{K\top} \quad \boldsymbol{\lambda}_i^{C\top}]^\top \in \mathbb{R}^{C+K}$ to obtain both the label-weight and model-weight.

Table 2: Parameters for the outer-problems in Section 6

Network	Method	L2 reg. rate	Hyper-learning rate
FC and FixU	PDBO-DA	0	0.1
	PDBO-MTL	0.01	1.0
	PDBO-MTL&DA	0.01 for $\boldsymbol{\lambda}_i^K$ 0.0005 for $\boldsymbol{\lambda}_i^C$	1.0 for $\boldsymbol{\lambda}_i^K$ 0.1 for $\boldsymbol{\lambda}_i^C$
StoU and StoD	PDBO-DA	0	0.1
	PDBO-MTL	0.01	0.1
	PDBO-MTL&DA	0.01 for $\boldsymbol{\lambda}_i^K$ 0.0005 for $\boldsymbol{\lambda}_i^C$	0.1 for $\boldsymbol{\lambda}_i^K$ 0.1 for $\boldsymbol{\lambda}_i^C$

For all $i \in [n]$ in the outer-problem, we ran 20 outer-steps of Adam (Kingma & Ba, 2015) iterations with $(\beta_1, \beta_2) = (0.9, 0.999)$ from the initial hyperparameters $\mathbf{0}_C$, $\mathbf{0}_K$, and $\mathbf{0}_{C+K}$ for PDBO-DA, PDBO-MTL, and PDBO-MTL&DA, respectively. For Adam optimizer, we adopted different learning rate shown in Table 2 (Hyper-learning rate). We adopt HGP for FC and FixU setting, and VR-HGP with $(\alpha, \beta) = (0.9, 0.1)$ for StoU and StoD settings. Both HGP and VR-HGP ran $M = 200$ estimation steps using iteration Eq. (4b) in all the settings. We also made a practical modification in HGP to sample $\tilde{\boldsymbol{A}}^{(m)}$ and $\tilde{\boldsymbol{B}}^{(m)}$ together at the single m-th round, which leads the same length of the Neumann series with the half sampling costs of the original HGP, although they are no more unbiased. For all the approaches the cases and for all $i \in [n]$, we used the average cross-entropy loss over the local training dataset of the i-th node and L2 regularization loss of $\boldsymbol{\lambda}_i$ for F_i with the rates shown in Table 2 (L2 reg. rate). We reported the mean test accuracy of an intermediate step that had maximum validation accuracy (i.e. early stopping) which was sampled independently from the training dataset as described in appendix D.

Baseline approaches We compared our approaches with baselines for each communication setting.

For FC and FixU settings, we compared with several personalization approaches: a personalized model trained only on the local dataset (Local), FedAvg with local tuning (FedAvg+) (Jiang et al., 2019), Clustered-FL (Sattler et al., 2020), pFedMe (T Dinh et al., 2020), and centralized and decentralized version of FedEM adopted in Marfoq et al. (2021). We also trained global models using SGP (Nedić & Olshevsky, 2016; Assran et al., 2019) and FedProx (Li et al., 2020). From the fact that SGP recovers FedAvg and DSGD on FC and FixU, respectively, we treat them as equivalent approaches. All the approaches on FC and FixU followed the training procedure with epoch-wise communication in Marfoq et al. (2021) while using Eq. (4b) for HGP computation. And any method ran on StoU and StoD adopted the SGP iteration (Eq. (4b)) with $T = 600$ steps, batch size $|\zeta_i| = 128$, L2 regularization with 0.001 decay. For SGP StoU and StoD, we adopted the learning rate $\alpha_i = 0.05$ for SGP, Local, and PDBO-DA, $\alpha_i = 0.25$ for PDBO-MTL and PDBO-MTL&DA. Those learning rates were scheduled to be multiplied by 0.1 at $t = 500, 550$. As we have no baseline ensemble model approach (i.e. FedEM) to be compared to our PDBO-MTL and PDBO-MTL&DA in StoU and StoD, we also examined our performance improvement from the initial hyperparameter. We confirmed PDBO-MTL and PDBO-MTL&DA improved their test accuracy from the initial hyperparameter both in StoU and StoD, confirming the performance gain of PDBO-MTL and PDBO-MTL&DA from SGP were not solely due to their differences in architectures and learning rates.

Dataset and model We adopted the procedure of generating a federated version of EMNIST in Marfoq et al. (2021) except for training and validation split. In our experiments, we consider 10%of the EMNIST dataset as in that were partitioned according to Dirichlet allocation of parameter $\alpha = 0.4$ over $n = 100$ clients as in Marfoq et al. (2021). We randomly selected 20% of the obtained dataset to make a validation dataset. We use the validation dataset only for the early stopping in outer-optimization of PDBO-DA, PDBO-MTL, and PDBO-MTL&DA. We trained the same CNN in

Table 3: Comparison of the gradient-based PDBO, CDBO, and CDBO-Local.

Study	Bilevel problem	Communication network	Hyper-gradient	No $O(d_{\boldsymbol{x}} \times d_{\boldsymbol{\lambda}})$ and $O(d_{\boldsymbol{x}} \times d_{\boldsymbol{x}})$	
				in communication	in computation
Ours	PDBO	Stochastic directed	GlobalGrad	✓	✓
Chen et al. (2022)	CDBO	Static undirected	ClientGrad		
Gao et al. (2022)	CDBO	Static undirected	LocalGrad	✓	✓
Yang et al. (2022)	CDBO	Static undirected	GlobalGrad		
Tarzanagh et al. (2022)	CDBO	Centralized	GlobalGrad	✓	✓
Li et al. (2022)	CDBO-Local	Centralized	LocalGrad	✓	✓
Liu et al. (2022)	CDBO-Local	Static undirected	LocalGrad	✓	✓
Lu et al. (2022)	CDBO-Local	Static undirected	LocalGrad	✓	✓

Marfoq et al. (2021) for all the baselines with a single model and PDBO-DA, and for base-predictor of FedEM, PDBO-MTL, and PDBO-MTL&DA.

E GRADIENT-BASED DISTRIBUTED BILEVEL OPTIMIZATION

We compare concurrent studies of distributed bilevel optimization (Chen et al., 2022; Tarzanagh et al., 2022; Gao et al., 2022; Yang et al., 2022; Li et al., 2022; Liu et al., 2022; Lu et al., 2022) in terms of problem settings, applicability on communication networks, hyper-gradient value to estimate, and complexity in communication and computation.

Bilevel problem setting We categorize them into two problems (Bilevel problem in Table 3): the consensus distributed bilevel optimization (CDBO) (Chen et al., 2022; Tarzanagh et al., 2022; Gao et al., 2022; Yang et al., 2022) and CDBO with the local inner-problem (CDBO-Local) (Li et al., 2022; Liu et al., 2022; Lu et al., 2022).

CDBO pursue consensus also in outer-problem, which can be obtained by imposing $\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_j$ for all $i, j \in [n]$ on PDBO outer-problem (Eq. (5-left)):

$$\min_{\substack{\boldsymbol{\lambda}_i \\ \boldsymbol{\lambda}_i = \boldsymbol{\lambda}_j, \forall j}} \frac{1}{n} \sum_{i=1}^{n} F_i \left(\boldsymbol{x}_i^* \left(\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_n \right), \boldsymbol{\lambda}_i \right), \text{ s.t. } \boldsymbol{x}_i^* = \operatorname*{arg\,min}_{\substack{\boldsymbol{x}_i \\ \boldsymbol{x}_i = \boldsymbol{x}_j, \forall j}} \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\xi_i} \left[f_i \left(\boldsymbol{x}_i, \boldsymbol{\lambda}_i; \xi_i \right) \right], \tag{38}$$

Chen et al. (2022); Tarzanagh et al. (2022); Gao et al. (2022); Yang et al. (2022) applied CDBO to hyper-parameter (e.g. L2 regularization coefficient) optimization.

While CDBO-Local also requires consensus in the outer-problem as in CDBO, its inner-problem is a local optimization problem in which optimal parameters are independent of each other client, unlike PDBO and CDBO:

$$\min_{\substack{\boldsymbol{\lambda}_i \\ \boldsymbol{\lambda}_i = \boldsymbol{\lambda}_j, \forall j}} \frac{1}{n} \sum_{i=1}^{n} F_i \left(\boldsymbol{x}_i^* \left(\boldsymbol{\lambda}_i \right), \boldsymbol{\lambda}_i \right), \text{ s.t. } \boldsymbol{x}_i^* = \operatorname*{arg\,min}_{\boldsymbol{x}_i} \mathbb{E}_{\xi_i} \left[f_i \left(\boldsymbol{x}_i, \boldsymbol{\lambda}_i; \xi_i \right) \right], \tag{39}$$

Lu et al. (2022) demonstrated the ability of CDBO-Local problem to handle personalization tasks. However, no client in CDBO-Local can benefit from the others in the inner loop for better generalization. We note that in our PDBO, both outer and inner problems are optimized from the global information; the inner parameter is trained for consensus among the clients and the outer parameter is optimized to improve the total performance across all the clients.

Communication networks The communication networks can be categorized into stochastic directed, static undirected, and centralized (Communication network in Table 3).

Studies for CDBO (Chen et al., 2022; Gao et al., 2022; Yang et al., 2022) and CDBO-Local (Liu et al., 2022; Lu et al., 2022) suppose the communication networks are static and undirected. More specifically, they assume the weighted mixing matrix $\boldsymbol{P}^{(t)}$ to be a double-stochastic matrix at all time steps $t \in \mathbb{N}$ for the consensus of DSGD in the outer-problem (Liu et al., 2022; Lu et al., 2022) (i.e. $\boldsymbol{x}_i = \boldsymbol{x}_j, \forall i, j \in [n]$), and both in the outer-problem and inner-problem (Chen et al., 2022; Gao et al., 2022; Yang et al., 2022) (i.e. $\boldsymbol{x}_i = \boldsymbol{x}_j, \boldsymbol{\lambda}_i = \boldsymbol{\lambda}_j, \forall i, j \in [n]$).

Tarzanagh et al. (2022); Liu et al. (2022) addresses the consensus in the outer-problem by adopting centralized communication settings so that the single global hyperparameter are shared among the clients at every step.

Our HGP is the only method that runs even on stochastic and directed communication networks.

In terms of the consensus, we can relax the assumption of the static undirected communication in Chen et al. (2022); Gao et al. (2022); Yang et al. (2022); Liu et al. (2022); Lu et al. (2022) to the stochastic and directed networks by replacing DSGD with SGP for the inner-loop and outer-loop. However, in terms of the hyper-gradient estimation, we cannot naively replace the communication networks setting as discussed in Section 4.2.

Hyper-gradient to estimate Both PDBO and CDBO require hyper-gradient estimation as they involve the interaction of clients in the inner-problem. However, the estimated hyper-gradient varies among the studies, so we categorize them into GlobalGrad, ClientGrad, and LocalGrad (Hyper-gradient in Table 3). Our HGP and Yang et al. (2022); Tarzanagh et al. (2022) aim at estimating the gradient of the average outer-objective across the client with respect to the hyperparameter of the client (GlobalGrad), i.e. $\mathrm{d}_{\boldsymbol{\lambda}_i} \bar{F}\left(\boldsymbol{x}^*\left(\boldsymbol{\lambda}\right), \boldsymbol{\lambda}\right) \in \mathbb{R}^{d_{\boldsymbol{\lambda}}}$.

Chen et al. (2022) estimate slightly different hyper-gradient, that is, gradient of client outer-objective with respect to the hyperparameter of the client (ClientGrad), i.e. $\mathrm{d}_{\boldsymbol{\lambda}_i} F_i\left(\boldsymbol{x}_i^*\left(\boldsymbol{\lambda}\right), \boldsymbol{\lambda}_i\right)$. Unlike GlobalGrad, ClientGrad only lets the client know how the perturbation on the client's hyperparameter changes its own outer-objective. Thus the gradient step of the client hyperparameter using ClientGrad is not supposed to improve the performance of the others, which is not the case with GlobalGrad.

Gao et al. (2022) estimates the LocalGrad which is equivalent to the hyper-gradient estimation of SGD that estimates $\mathrm{d}_{\boldsymbol{\lambda}_i} F_i\left(\boldsymbol{x}_i^*\left(\boldsymbol{\lambda}_i\right), \boldsymbol{\lambda}_i\right)$. LocalGrad differs from ClientGrad because LocalGrad needs no communication because the optimal inner parameter $\boldsymbol{x}_i^*$ is only parameterized by its hyperparameter $\boldsymbol{\lambda}_i$.

Complexity in communication and computation For a fair comparison, we compare the complexity of communication and computation between methods that intend to estimate the same hyper-gradient. Note that we only focus on the requirement of computation or communication for the full Jacobian matrix as it is dominant in decentralized hyper-gradient estimation (rightmost two columns of Table 3).

No approach for LocalGrad involves the full Jacobian computation and communication as they can naively adopt efficient algorithms such as backward mode. For GlobalGrad, the algorithm proposed by Yang et al. (2022) is complex both in computation and communication as they involve computations and communications of full Jacobian matrix ($O\left(d_{\boldsymbol{y}} \times d_{\boldsymbol{\lambda}}\right)$) and Hessian matrix ($O\left(d_{\boldsymbol{y}} \times d_{\boldsymbol{y}}\right)$). Tarzanagh et al. (2022) and our HGP enjoys reasonable complexity because these methods avoid computation and communication of full Jacobian by using Jacobian-vector products.