Supplementary Materials for HFedATM: Hierarchical Federated Domain Generalization via Optimal Transport and Regularized Mean Aggregation

Thinh Nguyen[1, 2], Trung Phan[2], Binh T. Nguyen[3], Khoa D Doan[1, 2], Kok-Seng Wong[1, 2*]
[1]VinUni-Illinois Smart Health Center, VinUniversity, Hanoi, Vietnam
[2]College of Engineering & Computer Science, VinUniversity, Hanoi, Vietnam
[3]Vietnam National University - Ho Chi Minh city, University of Science, Vietnam
{thinh.nth,23trung.pl}@vinuni.edu.vn, ngtbinh@hcmus.edu.vn, {khoa.dd,wong.ks}@vinuni.edu.vn

A. Detailed Theoretical Analysis

A.1. Generalization-error Bound

We firstly need the following definitions and assumptions:

Definition 1. *Let H be any hypothesis class whose loss $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ is L-Lipschitz and γ-Holder-continuous. For two marginal distributions P_X, Q_X we denote the H-divergence by*

$$d_H(P_X, Q_X) = 2 \sup_{h \in H} \left| \Pr_{P_X}[h(x) = 1] - \Pr_{Q_X}[h(x) = 1] \right|.$$

Assumption 1 (Bounded and Lipschitz Loss)**.** *The loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ is bounded and L-Lipschitz continuous in its first argument, meaning there exists a constant $L > 0$ such that for all y, u, v:*

$$|\ell(u, y) - \ell(v, y) \leq L\|u - v\|.$$

Assumption 2 (Holder Continuity)**.** *The loss function $\ell(f(u), y)$ is γ-Holder continuous in its input with constant L_γ, i.e., there exists $\gamma \in (0, 1]$ such that for all y, u, v:*

$$|\ell(f(u), y) - \ell(f(v), y)| \leq L_\gamma \|u - v\|^\gamma.$$

Definition 2. *Let $E = \{1, \ldots, N_E\}$ be the set of stations and $C_e = \{1, \ldots, N_e\}$ the clients managed by the station e. At each round, a subset $\hat{C}_e \subseteq C_e$ of active clients is selected. Let $S_{e,i} = \{(x_{e,i}^{(j)}, y_{e,i}^{(j)})\}_{j=1}^{n_{e,i}} \sim P_{XY}^{e,i}$ denote the client-level source domains, and these distributions are typically heterogeneous (domain shift). The stations do not hold or access any raw data. Instead, they aggregate models trained by their associated clients. Let $P_{XY}^\star$ be an unseen target domain such that $P_{XY}^\star \neq P_{XY}^{e,i} \forall e, i$. The HFedDG task seeks a hypothesis h that minimizes the expected loss $D_{\text{target}}(h) = \mathbb{E}_{(x,y)\sim P_{XY}^\star}[\ell(h(x), y)]$ without exposing client's data.*

We prove the Thereom 1 as follows:

Theorem 1 (Generalization-error bound for HFedDG)**.** *Let $d_{\mathcal{H}}$ be the $\mathcal{H}$-divergence. Given the client-level distributions $P_X^{e,i}$, let $\eta_e = \min_{i \in C_e} d_H(P_X^\star, P_X^{e,i})$ be the inner divergence and $\omega_e = \max_{i,j \in C_e} d_H(P_X^{e,i}, P_X^{e,j})$ be the breadth at each station e. Similarly, we define the server-level inner divergence and breadth as $\eta = \min_{e \in E, i \in C_e} d_H(P_X^\star, P_X^{e,i})$ and $\omega = \max_{(e,i),(e',j)} d_H(P_X^{e,i}, P_X^{e',j})$. Then, for any hypothesis $h \in \mathcal{H}$,*

$$\begin{aligned} D_{\text{target}}(h) \leq & \sum_{e=1}^{N_E} \sum_{i=1}^{N_e} \rho_{e,i}^\star D_{e,i}(h) + \frac{1}{2} \sum_{e=1}^{N_E} \rho_e^\star \omega_e \\ & + \frac{1}{2}\omega + \sum_{e=1}^{N_E} \rho_e^\star \eta_e + \eta + \lambda_{\mathcal{H}}(P_X^\star, P_X^\dagger). \end{aligned} \tag{1}$$

where $D_{e,i}(h) = \mathbb{E}_{(x,y)\sim P_{XY}^{e,i}}[\ell(h(x), y)]$ is the client-level risk, $\rho_{e,i}^\star \geq 0$ and $\rho_e^\star = \sum_{i \in C_e} \rho_{e,i}^\star \geq 0$, with $\sum_e \rho_e^\star = 1$, represent optimal distributional proximity from client distributions to the target distribution, $P_X^\dagger$ is the closest client-level distribution to $P_X^\star$, and $\lambda_{\mathcal{H}}$ is the joint-error term.

Proof. For every client distribution $P_{e,i}$, using the work of Ben-David et al. [4], we have

$$D_{\text{target}}(h) \leq D_{e,i}(h) + \frac{1}{2} d_{H\Delta H}(P_{e,i}, P^\star) + \lambda_H(P_{e,i}, P^\star).$$

Because $\lambda_H(P_{e,i}, P^\star) \leq \lambda_H(P^\dagger, P^\star)$ (choose the same optimal h), we replace the last term by $\lambda_H(P^\dagger, P^\star)$ for all clients. Multiply the above inequality by $\rho_{e,i}^\star$ and sum over

*Corresponding author: wong.ks@vinuni.edu.vn

(e,i), we have:

$$D_{\text{target}}(h) \le \sum_{e,i} \rho^\star_{e,i} D_{e,i}(h) + \frac{1}{2}\sum_{e,i} \rho^\star_{e,i} d_{H\Delta H}(P_{e,i}, P^\star) + \lambda_H(P^\dagger, P^\star). \tag{2}$$

For every station e, choose a pivot client $i^\dagger(e) = \arg\min_{j\in\mathcal{C}_e} d_H(P^\star, P_{e,j})$ and define the pivot distribution $P_e^\dagger \equiv P_{e,i^\dagger(e)}$. We further define the server pivot $P_X^\dagger \equiv \arg\min_{P\in\{P_e^\dagger\}} \sum_e \rho_e^\star d_H(P, P^\star)$. For any client (e,i),

$$d_{H\Delta H}(P_{e,i}, P^\star) \le d_H(P_{e,i}, P_e^\dagger) + d_H(P_e^\dagger, P^\dagger) + d_H(P^\dagger, P^\star). \tag{3}$$

This is because the $H\Delta H$-divergence upper-bounds the ordinary H-divergence, and the latter obeys the triangle inequality. Recall that we have the intra-station term $d_H(P_{e,i}, P_e^\dagger) \le \omega_e$, the station-server term $d_H(P_e^\dagger, P^\dagger) \le \omega$, and the server-target term $d_H(P^\dagger, P^\star) = \eta$. Similarly, $d_H(P_e^\dagger, P^\star) = \eta_e$. Plugging these into Equation 3 and then into Equation 2, we have

$$L_{\text{target}}(h) \le \sum_{e,i} \rho^\star_{e,i} L_{e,i}(h) + \frac{1}{2}\Big[\underbrace{\eta + \sum_e \rho_e^\star \eta_e}_{\text{inner divergences}} + \underbrace{\omega + \sum_e \rho_e^\star \omega_e}_{\text{breadths}}\Big] + \lambda_H(P^\dagger, P^\star).$$

We have proved the theorem. □

A.2. HFedATM's Convergence

Building upon the HFedDG error bound, we propose the next lemma, sharpening our theoretical analysis by employing Holder continuity:

Lemma 1. *Under Assumptions 1 and 2, for any measurable f and distributions P, Q:*

$$\big|\mathbb{E}_P[f] - \mathbb{E}_Q[f]\big| \le \tfrac{1}{2} L^\gamma d_H(P,Q)^\gamma.$$

Proof. Let $h^\star = \arg\max_{h\in H} \big|\Pr_P[h] - \Pr_Q[h]\big|$. By Hölder continuity, $|f(u) - f(v)| \le L\|u - v\|^\gamma$. Applying the variational definition of d_H and Jensen's inequality gives

$$\begin{aligned}\big|\mathbb{E}_P[f] - \mathbb{E}_Q[f]\big| &\le \mathbb{E}_{\substack{x\sim P\\ x'\sim Q}} \big|f(h^\star(x)) - f(h^\star(x'))\big| \\ &\le L\mathbb{E}\|h^\star(x) - h^\star(x')\|^\gamma \\ &\le \tfrac{1}{2} L^\gamma d_H(P,Q)^\gamma.\end{aligned}$$

We have proved the lemma. □

Next, since our framework employs FOT Alignment, it is crucial to confirm that the alignment is invariant under permutation. The following lemma verifies this property.

Lemma 2. *Let $D_{OT}(W_1, W_2) = \min_{\Pi\in\mathcal{U}} \langle \Pi, C\rangle$. For any permutation matrix P,*

$$D_{\text{OT}}(PW_1, PW_2) = D_{\text{OT}}(W_1, W_2).$$

Proof. Premultiply W_1 and W_2 by the same permutation matrix P. The new cost matrix becomes

$$\begin{aligned}C'_{ab} &= \big\|(PW_1)_a - (PW_2)_b\big\|_2^2 \\ &= \big\|w_{1,P^{-1}(a)} - w_{2,P^{-1}(b)}\big\|_2^2 \\ &= C_{P^{-1}(a)P^{-1}(b)}.\end{aligned}$$

For any admissible plan $\Pi \in \mathcal{U}$ define

$$\Pi'_{ab} = \Pi_{P^{-1}(a)P^{-1}(b)}.$$

Because rows and columns are merely re-indexed, Π' satisfies the same marginals $\Pi'\mathbf{1} = \frac{1}{c}\mathbf{1}$ and $(\Pi')^\top\mathbf{1} = \frac{1}{c}\mathbf{1}$; hence $\Pi' \in \mathcal{U}$. Moreover, the mapping $\Pi \mapsto \Pi'$ is a bijection of $\mathcal{U}$. Using the change of indices,

$$\begin{aligned}\langle \Pi', C'\rangle &= \sum_{a,b} \Pi_{P^{-1}(a)P^{-1}(b)} C_{P^{-1}(a)P^{-1}(b)} \\ &= \sum_{u,v} \Pi_{uv} C_{uv} = \langle \Pi, C\rangle.\end{aligned}$$

Because every feasible Π for (W_1, W_2) maps to a feasible Π' for (PW_1, PW_2) with identical cost, the minimal costs coincide:

$$\begin{aligned}D_{\text{OT}}(PW_1, PW_2) &= \min_{\Pi'\in\mathcal{U}} \langle \Pi', C'\rangle \\ &= \min_{\Pi\in\mathcal{U}} \langle \Pi, C\rangle \\ &= D_{\text{OT}}(W_1, W_2).\end{aligned}$$

We have proved the lemma. □

Integrating previous lemmas, we show that HFedATM's design choices lead to a tighter error bound:

Theorem 2. *Assume the local training at every client achieves $D_{e,i}(h^{(R)}) \le \varepsilon_{\text{local}}$ for all (e,i), and that the conditions of Theorem 1 hold. Then the hypothesis $h_{\text{ATM}}^{(R)}$ output by HFedATM satisfies*

$$\begin{aligned}D_{\text{target}}\big(h_{\text{ATM}}^{(R)}\big) \le& \varepsilon_{\text{local}} + \tfrac{1}{2}(1-\beta)^R(\omega^{(0)} + \sum_{e=1}^{N_E} \rho_e^\star \omega_e^{(0)}) \\ &+ (1-\beta)^R(\eta^{(0)} + \sum_{e=1}^{N_E} \rho_e^\star \eta_e^{(0)}) \\ &+ \lambda_{\mathcal{H}}(P_X^\star, P_X^\dagger),\end{aligned}$$

where the weights $\{\rho_e^\star\}$ are the optimal coupling weights of Theorem 1.

Algorithm 1: HFedATM: HFedDG via OptimAl Transport and regularized Mean aggregation

Input : global rounds R, station rounds N, client epochs E, learning rate η, shrinkage α, OT regularizer ε.
Output: global model $h_{\text{ATM}}^{(R)}$.

for $r \leftarrow 1$ **to** R **do** // server initialization
  // **Step 0: broadcast**
  The server multicasts previous global model $h^{(r-1)}$ to every station e.
  // **Step 1: client training**
  foreach *station* e **do**
    foreach $n \leftarrow 1$ **to** N **do**
      Select client subset $\hat{C}_e$.
      foreach $i \in \hat{C}_e$ **do**
        Pull $h^{(r-1)}$ and train E epochs with any FedDG algorithm.
        Send $\theta_{e,i}$ to station e.
        if $n == N$ **then**
          Each $l \in \mathcal{L}_{\text{lin}}$, calculate Gram $G_{e,i}^{(l)}$ and send to e.
  // **Step 2: station aggregation**
  foreach *station* e **do**
    Aggregate $\{\theta_{e,i}\}$ with FedAvg or FedDG approaches $\rightarrow$ station model h_e.
    Get station Gram $G_e^{(l)} \leftarrow \frac{1}{|\mathcal{S}_e|}\sum_i G_{e,i}^{(l)}$.
    Shrink $\widehat{G}_e^{(l)} \leftarrow \alpha G_e^{(l)} + (1-\alpha)\,\text{diag}(G_e^{(l)})$.
    Send $(h_e, \widehat{G}_e^{(l)})$ to the server.
  // **Step 3: server FOT Alignment**
  foreach $l \in \mathcal{L}_{\text{conv}}$ **do**
    Station 1 is reference.
    for $e = 2$ **to** N_E **do**
      Calculate index $\Pi_{1,e}^{(l)}$ and send back to e.
  // **Step 4: model merging**
  foreach $l \in \mathcal{L}_{\text{conv}}$ **do**
    $\overline{W}^{(l)} \leftarrow \dfrac{\sum_e \gamma_e W_e^{(l)}}{\sum_e \gamma_e}$ with $\gamma_e = |\mathcal{S}_e|$.
  foreach $l \in \mathcal{L}_{\text{lin}}$ **do**
    $W_{\text{ATM}}^{(l)} \leftarrow \big(\sum_e \widehat{G}_e^{(l)}\big)^{-1}\big(\sum_e \widehat{G}_e^{(l)}\widetilde{W}_e^{(l)}\big)$.
  // **Step 5: assemble & broadcast**
  Assemble $h_{\text{ATM}}^{(r)}$ from $\{\overline{W}^{(l)}\}_{\text{conv}}$ and $\{W_{\text{ATM}}^{(l)}\}_{\text{lin}}$ and broadcast $h_{\text{ATM}}^{(r)}$ to stations.

Proof. Let $\omega^{(r-1)}$ and $\omega_e^{(r-1)}$ be the breadths before FOT in round r. We have that FOT finds the optimal permutation $\Pi^\star$ minimizing entropic-OT cost

$$\langle \Pi^\star, C\rangle = \min_{\Pi\in\mathcal{U}}\langle \Pi, C\rangle \le \tfrac{1}{c}\omega^{(r-1)}B^2$$

(bounded because features are bounded by B). Let $\beta_{\text{FOT}} := \frac{\langle \Pi^\star, C\rangle}{\langle I, C\rangle} \in (0,1)$. Then, after permuting filters identically at all stations,

$$\omega^{(r)} = (1-\beta_{\text{FOT}})\omega^{(r-1)}, \quad \omega_e^{(r)} = (1-\beta_{\text{FOT}})\omega_e^{(r-1)}, \tag{4}$$

where Lemma 2 guarantees the contraction is permutation-invariant. We recall that RegMean solves the closed-form minimizer.

$$W^{(r)} = \alpha\bar{W}^{(r-1)} + (1-\alpha)W^{(r-1)},$$

where $\bar{W}^{(r-1)}$ is the average filter after FOT and $\alpha \in (0,1)$. Because ℓ is L-Lipschitz and Holder-continuous, the change in feature space distance contracts by at least the factor $1-\alpha$:

$$\eta^{(r)} = (1-\alpha)\eta^{(r-1)}, \quad \eta_e^{(r)} = (1-\alpha)\eta_e^{(r-1)}. \tag{5}$$

Let $\beta := 1-(1-\beta_{\text{FOT}})(1-\alpha)$. Then both breadth and divergence terms satisfy

$$\omega^{(r)} \le (1-\beta)\omega^{(r-1)}, \quad \eta^{(r)} \le (1-\beta)\eta^{(r-1)}. \tag{6}$$

Because $\beta_{\text{FOT}} > 0$, we have $0 < \beta < 1$. Starting from initial values at $r = 0$ and unrolling Equation 6 for R rounds gives

$$\eta^{(R)} \le (1-\beta)^R\eta^{(0)}, \quad \omega^{(R)} \le (1-\beta)^R\omega^{(0)},$$

and likewise for $\eta_e^{(R)}, \omega_e^{(R)}$. Insert these into Theorem 1:

$$\begin{aligned} D_{\text{target}}\big(h^{(R)}\big) \le \varepsilon_{\text{local}} + \frac{1}{2}\Big[&(1-\beta)^R\big(\eta^{(0)} + \omega^{(0)}\big) \\ &+ (1-\beta)^R\sum_e \rho_e^\star\big(\eta_e^{(0)} + \omega_e^{(0)}\big)\Big] \\ &+ \lambda_H(P_X^\star, P_X^\dagger). \end{aligned}$$

We complete the proof. □

B. Integrated Algorithm

The complete HFedATM workflow couples the three computation tiers: clients, stations, and the server, into a single synchronous loop. Algorithm 1 displays the procedure; the subsequent paragraphs clarify what is computed at each tier, when communication occurs, and how privacy is preserved.

Data	Model	E	N	B	S	C/S	Opt	LR_0	Mom	WD	Sched	λ_{OT}	α	n_{iter}	Rounds
PACS	LeNet-5	10	5	32	10	10	SGD	0.01	0.00	0.00	cosine	0.05	0.75	25	200
	ResNet-18	10	5	32	10	10	SGD	0.01	0.00	0.00	cosine	0.05	0.75	25	200
	VGG-11	10	5	32	10	10	SGD	0.01	0.00	0.00	cosine	0.05	0.75	25	200
Office-Home	LeNet-5	10	5	32	10	10	SGD	0.01	0.00	0.00	cosine	0.05	0.75	25	200
	ResNet-18	10	5	32	10	10	SGD	0.01	0.00	0.00	cosine	0.05	0.75	25	200
	VGG-11	10	5	32	10	10	SGD	0.01	0.00	0.00	cosine	0.05	0.75	25	200
TerraInc	LeNet-5	10	5	32	10	10	SGD	0.01	0.00	0.00	cosine	0.05	0.75	25	200
	ResNet-18	10	5	32	10	10	SGD	0.01	0.00	0.00	cosine	0.05	0.75	25	200
	VGG-11	10	5	32	10	10	SGD	0.01	0.00	0.00	cosine	0.05	0.75	25	200
Amazon Reviews	RoBERTa-base	10	5	32	10	10	AdamW	3×10^{-5}	0.90	10^{-2}	cosine	0.05	0.75	25	200
	DeBERTa-base	10	5	32	10	10	AdamW	3×10^{-5}	0.90	10^{-2}	cosine	0.05	0.75	25	200

Table 1. Hyper-parameters and implementation details used in our experiments.

C. Detailed Experimental Setup

Baseline Methods To comprehensively evaluate the performance of our proposed HFedATM method, we consider baseline approaches structured according to the two-stage HFL scenario. **At the client-station communication**, we first adopt FedAvg [12] as our baseline since it is the standard aggregation protocol in FL. To directly address the challenge of data heterogeneity, we further incorporate FedProx [10] which introduces a proximal regularization term into the local training objective, constraining client model updates and ensuring a more stable convergence under heterogeneous data conditions. Additionally, to assess DG capabilities, we select two FedDG baselines representing different categories: FedSR [13], a regularization-based technique, and FedIIR [5], an aggregation-based method. FedSR applies feature-level regularization, promoting simplicity and domain-invariant representations across multiple clients, while FedIIR performs gradient alignment across client models, facilitating invariant feature learning robust against distributional shifts. **At the station-server communication**, we use standard weight averaging (Avg) as our baseline aggregation method. All baseline implementations and comparisons were conducted using the established FL framework provided by Tan et al. [15].

Datasets Our experiments were conducted across diverse datasets commonly employed in DG studies: **PACS** [9], featuring stylistically varied images across photo, art-painting, cartoon, and sketch domains; **Office-Home** [16], comprising images across art, clipart, product, and real-world domains with a rich class diversity; **TerraInc** [3], containing camera-trap images from distinct wildlife locations; and **Amazon Reviews** [2], involving sentiment classification across distinct product categories. This selection of datasets collectively provides comprehensive coverage of different data types, domains, and challenges to robustly evaluate our proposed method.

Heterogeneous Partitioning To synthesize controllable non-IID data distributions across clients, we adopt the Heterogeneous Partitioning strategy proposed by Bai et al. [1]. Specifically, given D domains (or classes) and C clients, the algorithm first assigns a subset of domain indices D_c to one or more clients, subsequently allocating samples to each client-domain pair (d, c) according to:

$$n_{d,c}(\lambda) = \lambda \frac{n_d}{C} + (1-\lambda) \frac{\mathbf{1}[d \in D_c]\, n_d}{|\{c' : d \in D_{c'}\}|}, \qquad (7)$$

where $\lambda \in [0, 1]$, and n_d denotes the total number of samples in domain d. The parameter λ controls the degree of heterogeneity: a value of $\lambda = 1.0$ corresponds to an IID distribution, where each client receives data proportionally from all domains; conversely, $\lambda = 0.0$ results in maximum heterogeneity, assigning each client exclusively to its designated domains. Intermediate values (e.g., $\lambda = 0.1$) smoothly interpolate between these extremes, controlling domain-level imbalance.

Model Architectures For vision datasets, we employed LeNet-5 [8] (a convolutional neural network with 7 layers, consisting of two convolutional layers followed by pooling operations and three fully-connected layers, totaling approximately 60,000 parameters), ResNet-18 [6] (an 18-layer residual network architecture with convolutional and identity skip-connections, comprising around 11 million parameters), and VGG11 [14] (an 11-layer convolutional neural network featuring eight convolutional layers followed by three fully-connected layers, amounting to approximately 133 million parameters). For NLP tasks, we utilized transformer-based language models, including RoBERTa-base [11] (Robustly Optimized

BERT Pre-training Approach, a 12-layer transformer encoder with around 125 million parameters, known for improved masked-language modeling through enhanced training strategies) and DeBERTa-base [7] (Decoding-enhanced BERT with disentangled attention, a 12-layer transformer encoder architecture comprising roughly 140 million parameters, notable for disentangling content and position representations within its attention mechanisms, significantly boosting model performance).

Hyper-parameter Tuning All hyper-parameters, fixed constants, and runtime configurations used in our experiments are detailed in Table 1, with corresponding descriptions provided in Table 2. We conducted experiments on NVIDIA RTX 3090 GPUs, using three random seeds $\{0, 1, 2\}$) to ensure reproducibility.

Group	Column	Meaning
Data/Model	Data	Dataset name
	Model	Backbone architecture
Federation	E	Local epochs per station round
	N	Station rounds per server round
	B	Batch size
	S	Number of stations
	C/S	Clients per station
Optimization	Opt	Optimizer (SGD/AdamW)
	LR_0	Initial learning rate
	Mom	Momentum or β_1
	WD	Weight decay
	Sched	Learning rate scheduler
HFedATM-specific	λ_{OT}	Sinkhorn regularizer
	α	RegMean shrinkage parameter
	n_{iter}	Sinkhorn iterations
Runtime	Rounds	Global rounds

Table 2. Legend of each hyper-parameter.

D. Sensitivity Analysis

We study how HFedATM behaves under changes in the hierarchical configuration. Results are averaged over three random seeds and all target domains. We report performance for all four client-station baselines and all datasets, always using HFedATM at the station-server level.

D.1. Effect of Hierarchical Schedule

We first investigate how HFedATM reacts to changes in the schedule, controlled by the number of station rounds per server round and the total number of server rounds.

Varying station rounds. We fix Rounds $= 200$, $S = 10$, $C/S = 10$, $E = 10$, and sweep $N \in \{1, 3, 5, 10\}$ while keeping all other hyper-parameters as in Table 1. Table 3 reports the average target-domain accuracy for all methods and datasets. Across all cases we see a consistent pattern. When $N = 1$, each station performs only one round of client aggregation before communicating with the server, which weakens within-station consensus and yields noticeably lower accuracy. Increasing to a small number of station rounds ($N = 3$ or $N = 5$) recovers most of the benefits. Larger values ($N = 10$) bring at most marginal gains and sometimes a slight drop, while increasing the total amount of client-station computation. This confirms that HFedATM does not require many station rounds to be effective; the default value $N = 5$ used in the main experiments lies in a stable regime.

Dataset	Station Rounds	FedAvg	FedProx	FedSR	FedIIR
PACS	1	76.6	77.2	81.5	82.6
	3	78.0	78.7	83.4	84.2
	5	**78.6**	**79.2**	**83.9**	**84.6**
	10	78.5	79.1	83.8	84.5
Office-Home	1	57.2	57.8	63.5	64.4
	3	58.8	59.4	65.0	65.9
	5	**59.4**	**60.0**	**65.7**	**66.6**
	10	59.3	59.9	65.6	66.5
TerraInc	1	40.3	41.6	45.8	46.6
	3	42.0	43.3	47.3	48.1
	5	**42.5**	**43.8**	**48.0**	**48.8**
	10	42.4	43.7	47.9	48.7
Amazon Review	1	76.5	77.4	79.8	79.6
	3	78.0	78.9	80.9	81.3
	5	**78.5**	**79.4**	**81.2**	**81.6**
	10	78.4	79.3	81.1	81.5

Table 3. Effect of the number of station rounds N (with Rounds $=$ 200 fixed) on all datasets, averaged over target domains. HFedATM is used at the station-server level for all methods. Accuracy improves when increasing N from 1 to 3-5 and then saturates.

Varying server rounds. We now fix $N = 5$, $S = 10$, $C/S = 10$, $E = 10$, and vary the total number of server rounds $\in \{100, 200, 400\}$, again keeping all other hyper-parameters identical to Table 1. Table 5 shows that additional global communication generally improves accuracy across all baselines and datasets, but with diminishing returns. Overall, HFedATM benefits from a reasonable number of global rounds to repeatedly align station models, but does not require very long training. The default schedule (Rounds $= 200$, $N = 5$) used in the main experiments lies near the knee of the curve across all datasets.

D.2. Effect of Stations and Clients per Station

We now vary the hierarchical topology while keeping the total number of clients fixed at $C = S \times (C/S) = 100$. Specifically, we consider $S \in \{5, 10, 20\}$ stations with

Dataset	S	C/S	FedAvg +Avg	FedAvg +HFedATM	FedProx +Avg	FedProx +HFedATM	FedSR +Avg	FedSR +HFedATM	FedIIR +Avg	FedIIR +HFedATM
PACS	5	20	78.3	79.0	79.0	79.6	82.0	84.1	82.7	85.0
	10	10	76.8	78.6	77.5	79.2	80.5	83.9	81.2	84.6
	20	5	75.1	78.2	75.8	78.8	78.8	83.5	79.5	84.2
Office-Home	5	20	59.3	59.8	60.3	60.4	63.7	66.1	64.3	67.0
	10	10	57.8	59.4	58.8	60.0	62.2	65.7	62.8	66.6
	20	5	56.1	59.0	57.1	59.6	60.5	65.3	61.1	66.2
TerraInc	5	20	43.7	42.9	44.5	44.2	46.0	48.4	46.5	49.2
	10	10	42.2	42.5	43.0	43.8	44.5	48.0	45.0	48.8
	20	5	40.5	42.1	41.3	43.4	42.8	47.6	43.3	48.4
Amazon Review	5	20	71.4	78.9	72.4	79.8	73.8	81.9	73.7	82.0
	10	10	69.9	78.5	70.9	79.4	71.9	81.2	72.2	81.6
	20	5	68.2	78.1	69.2	79.0	70.1	80.7	70.5	81.2

Table 4. Effect of the number of stations S and clients per station C/S on all datasets, averaged over target domains, with total clients fixed at $C = 100$. HFedATM consistently improves all baselines and its gain increases when S is large and stations are more heterogeneous.

Dataset	Rounds	FedAvg	FedProx	FedSR	FedIIR
PACS	100	77.7	78.3	82.1	83.1
	200	**78.6**	**79.2**	**83.9**	**84.6**
	400	78.9	79.5	84.0	84.9
Office-Home	100	58.4	59.0	64.2	65.1
	200	**59.4**	**60.0**	**65.7**	**66.6**
	400	59.7	60.3	66.0	66.9
TerraInc	100	41.6	42.9	46.5	47.3
	200	**42.5**	**43.8**	**48.0**	**48.8**
	400	42.8	44.1	48.3	49.1
Amazon Review	100	77.5	78.4	80.3	80.4
	200	**78.5**	**79.4**	**81.2**	**81.6**
	400	78.8	79.7	81.3	81.9

Table 5. Effect of the number of server rounds (Rounds) on all datasets, with $N = 5$ fixed. More global rounds help HFedATM, but gains saturate around Rounds $= 200$, which is the value used in the main experiments.

$C/S \in \{20, 10, 5\}$ clients per station, respectively, and keep $E = 10$, $N = 5$, Rounds $= 200$, $B = 32$, and $\lambda = 1.0$ unchanged. For each configuration, we compare the standard station-server averaging (+Avg) with +HFedATM. Table 4 reports the average target-domain accuracy for all methods and datasets.

Two consistent trends emerge. First, HFedATM matches or improves the corresponding +Avg baseline for every topology and method. Second, as the number of stations S increases (and thus each station has fewer clients), inter-station heterogeneity becomes more pronounced: the +Avg baselines degrade, while HFedATM degrades much more slowly. This effect is particularly visible at $S = 20$, where HFedATM maintains high accuracy while +Avg suffers a clear drop, confirming that HFedATM is most beneficial when inter-station divergence is large.

E. Privacy Analysis

HFedATM aggregates only Gram matrices $G = X^\top X$ from client-side activations, inherently avoiding direct transmission of raw data or gradients. However, whether Gram matrices can leak sensitive information remains a concern. Here, we demonstrate theoretically that Gram matrices inherently protect against exact data inversion.

Theorem 3. *Given a Gram matrix $G = X^\top X \in \mathbb{R}^{m\times m}$, the activation matrix $X \in \mathbb{R}^{d\times m}$ cannot be uniquely recovered.*

Proof. Suppose X has rank $r \le \min(d, m)$. Consider any orthogonal matrix $Q \in \mathbb{R}^{m\times m}$ satisfying $Q^\top Q = I$. Define a new activation matrix as $\tilde{X} = XQ$. Then:

$$\tilde{X}^\top \tilde{X} = Q^\top X^\top X Q = Q^\top G Q.$$

If G has repeated eigenvalues or is rank-deficient (which is typically true since $d > m$), there exist infinitely many distinct matrices $\tilde{X}$ yielding the identical Gram matrix G. Therefore, the mapping from X to G is inherently many-to-one, ensuring non-invertibility and protecting against exact inversion attacks. □

References

[1] Ruqi Bai, Saurabh Bagchi, and David I Inouye. Benchmarking algorithms for federated domain generalization. *arXiv preprint arXiv:2307.04942*, 2023. 4

[2] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. *Advances in neural information processing systems*, 31, 2018. 4

[3] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018. 4

[4] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010. 1

[5] Yaming Guo, Kai Guo, Xiaofeng Cao, Tieru Wu, and Yi Chang. Out-of-distribution generalization of federated learning via implicit invariant relationships. In *International Conference on Machine Learning*, pages 11905–11933. PMLR, 2023. 4

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[7] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020. 5

[8] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 4

[9] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. 4

[10] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020. 4

[11] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 4

[12] 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*, pages 1273–1282. PMLR, 2017. 4

[13] A Tuan Nguyen, Philip Torr, and Ser Nam Lim. Fedsr: A simple and effective domain generalization method for federated learning. *Advances in Neural Information Processing Systems*, 35:38831–38843, 2022. 4

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[15] Jiahao Tan, Yipeng Zhou, Gang Liu, Jessie Hui Wang, and Shui Yu. pfedsim: Similarity-aware model aggregation towards personalized federated learning, 2023. 4

[16] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017. 4