

---

# Supplementary Material

---

## Appendices

An overview of the Appendix is provided below, covering algorithmic components, theoretical foundations, implementation specifics, extended empirical analysis, and broader contextual considerations.

|   |           |
|---|-----------|
| <b>A Hierarchical Adaptive Patch Mixup</b>  | <b>2</b>  |
| A.1 Hierarchical Random Partitioning of Spatial Patches . . . . .                   | 2         |
| A.2 Patch-wise Independent Interpolation and Fusion . . . . .                       | 2         |
| <b>B Pseudo Codes of FedKWAZ</b>  | <b>3</b>  |
| <b>C Theoretical Proofs</b>   | <b>4</b>  |
| C.1 Proof of Lemma 1 . . . . .  | 5         |
| C.2 Proof of Lemma 2 . . . . .  | 6         |
| C.3 Proof of Theorem 1 . . . . .  | 6         |
| C.4 Proof of Theorem 2 . . . . .  | 6         |
| <b>D Additional Experimental Details</b>  | <b>7</b>  |
| D.1 Experimental Environment . . . . .  | 7         |
| D.2 Datasets . . . . .  | 7         |
| D.3 Hyperparameter Settings . . . . .   | 7         |
| D.4 Detailed Setting of Model Heterogeneity . . . . .                               | 7         |
| <b>E Additional Experimental Results</b>  | <b>8</b>  |
| E.1 Impact of Feature Dimensions . . . . .  | 8         |
| E.2 Homogeneous Models Setting . . . . .  | 8         |
| E.3 Performance on Fashion-MNIST under Pathological and Dirichlet Data Settings . . | 9         |
| E.4 Performance under Low Client Participation Rates and High Client Drop Rates . . | 9         |
| E.5 Impact of SWAZ and DWAZ . . . . .   | 11        |
| E.6 Impact of Feature Shift . . . . .   | 11        |
| E.7 Effectiveness of HAPM vs. Hard Sample Selection . . . . .                       | 12        |
| E.8 Correlation between HAPM Parameters and Client Properties . . . . .             | 13        |
| E.9 Variance and Stability Analysis of Dynamically Selected HAPM Parameters . . . . | 14        |
| <b>F Visualizations of Data Distributions</b>                                       | <b>15</b> |
| <b>G Limitations</b>  | <b>17</b> |
| <b>H Broader Impacts</b>  | <b>17</b> |

## A Hierarchical Adaptive Patch Mixup

To explicitly capture knowledge discrepancies between heterogeneous models, a novel Hierarchical Adaptive Patch Mixup HAPM mechanism is introduced in FedKWAZ. High-quality samples with local diversity and hierarchical structure are generated through spatially granular partitioning and stochastic perturbation, supporting the identification and enhancement of weak knowledge-aware zones. The input to HAPM includes the images  $x$ , a mixing strength parameter  $\alpha$ , and a spatial granularity parameter  $G$ . Two key procedures are involved in HAPM: hierarchical random partitioning of spatial patches and independent interpolation of partitioned samples.

### A.1 Hierarchical Random Partitioning of Spatial Patches

To reflect knowledge differences across heterogeneous models at a local granularity, a spatially hierarchical random perturbation mechanism is incorporated into HAPM. The original input images  $X \in \mathbb{R}^{B \times C \times H \times W}$  (where  $B$ ,  $C$ ,  $H$ , and  $W$  denote the batch size, channels, height, and width) is partitioned into  $\sqrt{G} \times \sqrt{G}$  local patches. Each patch is then subjected to spatial perturbations to improve diversity and sensitivity to knowledge. Specifically, the base height and width of each patch are defined as:

$$h_{\text{base}} = \left\lfloor \frac{H}{\sqrt{G}} \right\rfloor, \quad w_{\text{base}} = \left\lfloor \frac{W}{\sqrt{G}} \right\rfloor \quad (1)$$

Next, the initial boundaries of each patch are constructed based on grid indices:

$$h_{\text{start}}^{(i,j)} = i \cdot h_{\text{base}}, \quad h_{\text{end}}^{(i,j)} = (i+1) \cdot h_{\text{base}}, \quad w_{\text{start}}^{(i,j)} = j \cdot w_{\text{base}}, \quad w_{\text{end}}^{(i,j)} = (j+1) \cdot w_{\text{base}} \quad (2)$$

Here, the grid indices  $i, j = 0, 1, \dots, \sqrt{G} - 1$  are used to specify patch positions. To increase the diversity of local perturbations, jitter factors  $\delta_h^{(i,j)}$  and  $\delta_w^{(i,j)}$  are introduced and sampled as integer variables from uniform distributions:

$$\delta_h^{(i,j)} \sim \mathcal{U}\left(-\frac{h_{\text{base}}}{4}, \frac{h_{\text{base}}}{4}\right), \quad \delta_w^{(i,j)} \sim \mathcal{U}\left(-\frac{w_{\text{base}}}{4}, \frac{w_{\text{base}}}{4}\right) \quad (3)$$

To ensure validity, the perturbed positions are clipped within image boundaries using the clip function:

$$\begin{aligned} \hat{h}_{\text{start}}^{(i,j)} &= \text{clip}\left(h_{\text{start}}^{(i,j)} + \delta_h^{(i,j)}, 0, H\right), \quad \hat{h}_{\text{end}}^{(i,j)} = \text{clip}\left(h_{\text{end}}^{(i,j)} + \delta_h^{(i,j)}, 0, H\right) \\ \hat{w}_{\text{start}}^{(i,j)} &= \text{clip}\left(w_{\text{start}}^{(i,j)} + \delta_w^{(i,j)}, 0, W\right), \quad \hat{w}_{\text{end}}^{(i,j)} = \text{clip}\left(w_{\text{end}}^{(i,j)} + \delta_w^{(i,j)}, 0, W\right) \end{aligned} \quad (4)$$

These designs jointly form the hierarchical structure of HAPM, where local-granularity patches are first divided in space and then perturbed at the patch level to enhance the diversity of the mixed samples.

### A.2 Patch-wise Independent Interpolation and Fusion

Based on the above spatial partitioning, the mixing coefficient  $\lambda^{(i,j)}$  is independently sampled for each patch zone from a Beta distribution:

$$\lambda^{(i,j)} \sim \text{Beta}(\alpha, \alpha), \quad \alpha > 0 \quad (5)$$

Given a randomly shuffled batch index  $\pi$ , the interpolation fusion is independently performed in each patch zone:

$$\begin{aligned} X^{\text{mix}}[:, :, \hat{h}_{\text{start}}^{(i,j)} : \hat{h}_{\text{end}}^{(i,j)}, \hat{w}_{\text{start}}^{(i,j)} : \hat{w}_{\text{end}}^{(i,j)}] &= \lambda^{(i,j)} X[:, :, \hat{h}_{\text{start}}^{(i,j)} : \hat{h}_{\text{end}}^{(i,j)}, \hat{w}_{\text{start}}^{(i,j)} : \hat{w}_{\text{end}}^{(i,j)}] \\ &\quad + (1 - \lambda^{(i,j)}) X[\pi, :, \hat{h}_{\text{start}}^{(i,j)} : \hat{h}_{\text{end}}^{(i,j)}, \hat{w}_{\text{start}}^{(i,j)} : \hat{w}_{\text{end}}^{(i,j)}] \end{aligned} \quad (6)$$

Through this mixing process, each local zone of the image is independently adjusted based on the patch-specific mixing strength, enabling HAPM to generate more diverse and knowledge-sensitive samples, thereby enhancing the model's recognition capacity in knowledge-weak zones.

## B Pseudo Codes of FedKWAZ

---

### Algorithm 1: FedKWAZ

---

**Input:**  $N$ , total number of clients;  $\rho$ , participation rate of clients in one round;  $T$ , total number of rounds;  $\eta$ , learning rate of private and proxy models;  $E_A, E_B$ , training epoch of the first and second stage;  $\alpha^*, \beta_1^*, \beta_2^*$ ,  $g^*$ , mixing parameters of HAPM;  $\tau$ , temperature of distillation;  $\mathcal{D}_k$ , the local dataset of the  $k$ -th client.

**Output:** Private and proxy models'  $P_c^{\mathcal{M}_k}, P_c^{\mathcal{Q}_k}$  and  $L_c^{\mathcal{M}_k}, L_c^{\mathcal{Q}_k}$ .

Randomly initialize the client local heterogeneous models  $[\mathcal{M}_1(\psi_1), \dots, \mathcal{M}_{N-1}(\psi_{N-1})]$  and local proxy homogeneous small models  $[\mathcal{Q}_1(\phi_1), \dots, \mathcal{Q}_{N-1}(\phi_{N-1})]$ .

**for each round**  $t=1, \dots, T-1$  **do**

**// Server Side:**

$S^t \leftarrow$  Randomly sample  $K$  clients from  $N$  clients; Broadcast  $\bar{P}_c^{\mathcal{M}}, \bar{P}_c^{\mathcal{Q}}, \bar{L}_c^{\mathcal{M}}, \bar{L}_c^{\mathcal{Q}}$  to them;

**for each client**  $k = 1, \dots, K$  **do**

        ClientUpdate( $\bar{P}_c^{\mathcal{M}}, \bar{P}_c^{\mathcal{Q}}, \bar{L}_c^{\mathcal{M}}, \bar{L}_c^{\mathcal{Q}}$ )

**end**

    Aggregate global semantic representation and decision output.

**// ClientUpdate:**

    Receive  $\bar{P}_c^{\mathcal{M}}, \bar{P}_c^{\mathcal{Q}}, \bar{L}_c^{\mathcal{M}}, \bar{L}_c^{\mathcal{Q}}$  from the server;

**for**  $k \in S^t$  **do**

**// Stage I: Global Knowledge Mutual Learning**

**for epoch**  $e = 1, \dots, E_A$  **do**

**for batch**  $(x, y) \in \mathcal{D}_k$  **do**

                Obtain representation and logits:

$z_x^{\mathcal{M}_k}, \hat{y}_x^{\mathcal{M}_k} \leftarrow \mathcal{M}_k(x; \psi_k), z_x^{\mathcal{Q}_k}, \hat{y}_x^{\mathcal{Q}_k} \leftarrow \mathcal{Q}_k(x; \phi_k)$ ;

                Compute global alignment loss:

$\ell_k^{\mathcal{M}_k \rightarrow \mathcal{Q}_k} = \ell_{CE}(\hat{y}_x^{\mathcal{M}_k}, y) + \ell_{CE}(\hat{y}_x^{\mathcal{M}_k}, \sigma(\bar{L}_c^{\mathcal{Q}})) + \|z_x^{\mathcal{M}_k} - \bar{P}_c^{\mathcal{Q}}\|_2^2$ ;

$\ell_k^{\mathcal{Q}_k \rightarrow \mathcal{M}_k} = \ell_{CE}(\hat{y}_x^{\mathcal{Q}_k}, y) + \ell_{CE}(\hat{y}_x^{\mathcal{Q}_k}, \sigma(\bar{L}_c^{\mathcal{M}})) + \|z_x^{\mathcal{Q}_k} - \bar{P}_c^{\mathcal{M}}\|_2^2$ ;

                Update:  $\psi_k^t \leftarrow \psi_k^{t-1} - \eta_\psi \nabla \ell_k^{\mathcal{M}_k \rightarrow \mathcal{Q}_k}, \phi_k^t \leftarrow \phi_k^{t-1} - \eta_\phi \nabla \ell_k^{\mathcal{Q}_k \rightarrow \mathcal{M}_k}$ ;

**end**

**end**

**// Stage II: Local KWAZ Mutual Learning**

**for epoch**  $e = 1, \dots, E_B$  **do**

**for batch**  $(x, y) \in \mathcal{D}_k$  **do**

                Compute base mutual learning losses:

$\ell_{\text{base}}^{\mathcal{M}_k \rightarrow \mathcal{Q}_k} = \text{KL} \left( \sigma(\hat{y}_x^{\mathcal{M}_k} / \tau) \parallel \sigma(\hat{y}_x^{\mathcal{Q}_k} / \tau) \right) \cdot \tau^2 + \|z_x^{\mathcal{M}_k} - z_x^{\mathcal{Q}_k}\|_2^2$ ;

$\ell_{\text{base}}^{\mathcal{Q}_k \rightarrow \mathcal{M}_k} = \text{KL} \left( \sigma(\hat{y}_x^{\mathcal{Q}_k} / \tau) \parallel \sigma(\hat{y}_x^{\mathcal{M}_k} / \tau) \right) \cdot \tau^2 + \|z_x^{\mathcal{Q}_k} - z_x^{\mathcal{M}_k}\|_2^2$ ;

                Generate SWAZ mixed samples:  $X_{\text{SWAZ}}^{\mathcal{M}_k \leftrightarrow \mathcal{Q}_k} = \text{HAPM}(x; \alpha^*, g^*)$ ;

                Compute semantic weak-awareness loss:

$\ell_{\text{SWAZ}}^{\mathcal{M}_k \rightarrow \mathcal{Q}_k} = \|z_{X_{\text{SWAZ}}^{\mathcal{M}_k \leftrightarrow \mathcal{Q}_k}}^{\mathcal{M}_k} - z_{X_{\text{SWAZ}}^{\mathcal{M}_k \leftrightarrow \mathcal{Q}_k}}^{\mathcal{Q}_k}\|_2^2 = \ell_{\text{SWAZ}}^{\mathcal{Q}_k \rightarrow \mathcal{M}_k}$ ;

                Generate DWAZ mixed samples:

$X_{\text{DWAZ}}^{\mathcal{M}_k \rightarrow \mathcal{Q}_k} = \text{HAPM}(x; \beta_1^*, g_1^*), X_{\text{DWAZ}}^{\mathcal{Q}_k \rightarrow \mathcal{M}_k} = \text{HAPM}(x; \beta_2^*, g_2^*)$ ;

                Compute decision weak-awareness losses:

$\ell_{\text{DWAZ}}^{\mathcal{M}_k \rightarrow \mathcal{Q}_k} = \text{KL} \left( \sigma(\hat{y}_{X_{\text{DWAZ}}^{\mathcal{M}_k \rightarrow \mathcal{Q}_k}}^{\mathcal{M}_k} / \tau) \parallel \sigma(\hat{y}_{X_{\text{DWAZ}}^{\mathcal{M}_k \rightarrow \mathcal{Q}_k}}^{\mathcal{Q}_k} / \tau) \right) \cdot \tau^2$ ;

$\ell_{\text{DWAZ}}^{\mathcal{Q}_k \rightarrow \mathcal{M}_k} = \text{KL} \left( \sigma(\hat{y}_{X_{\text{DWAZ}}^{\mathcal{Q}_k \rightarrow \mathcal{M}_k}}^{\mathcal{Q}_k} / \tau) \parallel \sigma(\hat{y}_{X_{\text{DWAZ}}^{\mathcal{Q}_k \rightarrow \mathcal{M}_k}}^{\mathcal{M}_k} / \tau) \right) \cdot \tau^2$ ;

                Update models again:

$\psi_k^t \leftarrow \psi_k^{t-1} - \eta_\psi \nabla [\ell_{\text{base}}^{\mathcal{M}_k \rightarrow \mathcal{Q}_k} + \ell_{\text{SWAZ}}^{\mathcal{M}_k \rightarrow \mathcal{Q}_k} + \ell_{\text{DWAZ}}^{\mathcal{M}_k \rightarrow \mathcal{Q}_k}]$ ;

$\phi_k^t \leftarrow \phi_k^{t-1} - \eta_\phi \nabla [\ell_{\text{base}}^{\mathcal{Q}_k \rightarrow \mathcal{M}_k} + \ell_{\text{SWAZ}}^{\mathcal{Q}_k \rightarrow \mathcal{M}_k} + \ell_{\text{DWAZ}}^{\mathcal{Q}_k \rightarrow \mathcal{M}_k}]$ ;

**end**

**end**

    Aggregate and upload updated  $P_c^{\mathcal{M}_k}, P_c^{\mathcal{Q}_k}, L_c^{\mathcal{M}_k}, L_c^{\mathcal{Q}_k}$  to the server.

**end**

**end**

---

## C Theoretical Proofs

The following notations and expressions are defined.  $t \in 0, 1, \dots, T-1$  is used to denote the  $t$ -th round of federated communication. The local loss function  $\mathcal{L}_k(w_k)$  is defined over the parameter set  $w_k$  of the local model  $\mathcal{M}_k$  and proxy model  $\mathcal{Q}_k$  on client  $k$ . In each communication round, a total of  $E = E_A + E_B$  local steps are conducted on client  $k$  in two stages. The start of the local update at communication round  $t$  is denoted as  $tE + 0$ , where the first local iteration of Stage I begins after the global semantic and decision anchors are received from the server. Stage I involves  $E_A$  steps of local update, with parameter sequence denoted as  $\{w_k^{tE+e}\}_{e=0}^{E_A}$ , and  $w_k^{tE+E_A}$  indicating the parameters at the end of this stage. Starting from  $w_k^{tE+E_A}$ , Stage II proceeds with  $E_B$  local update steps, with parameters denoted by  $\{w_k^{tE+e}\}_{e=E_A}^{E_A+E_B}$ , where  $w_k^{tE+E_A+E_B}$  represents the final parameters after the  $t$ -th local training round. After Stage II, the local representations and decision outputs are uploaded to the server for global aggregation and update, initiating the next communication round. The learning rates on client  $k$  are unified into a scalar  $\eta$ , comprising  $\eta_\psi$  and  $\eta_\phi$ .

**Assumption 1. Lipschitz Smoothness.** *The local loss function on any client  $k$  is assumed to satisfy the 1-Lipschitz smoothness condition. Specifically, for any parameter vectors  $w_k^{t_1}, w_k^{t_2}$ , it holds that:*

$$\|\nabla \mathcal{L}_k^{t_1}(w_k^{t_1}; x, y) - \nabla \mathcal{L}_k^{t_2}(w_k^{t_2}; x, y)\| \leq L_1 \|w_k^{t_1} - w_k^{t_2}\|, \forall t_1, t_2 > 0, (x, y) \in D_k \quad (7)$$

Moreover, this can be further expressed as:

$$\mathcal{L}_k^{t_1} - \mathcal{L}_k^{t_2} \leq \langle \nabla \mathcal{L}_k^{t_2}, (w_k^{t_1} - w_k^{t_2}) \rangle + \frac{L_1}{2} \|w_k^{t_1} - w_k^{t_2}\|^2 \quad (8)$$

**Assumption 2. Unbiased Gradient and Bounded Variance.** *On client  $k$ , during the  $t$ -th local update, the batch gradient sampled at  $w_k^t$  is denoted as  $g_{w,k}^t = \nabla \mathcal{L}_k^t(w_k^t; B_k^t)$ , and it is assumed to satisfy unbiasedness and bounded variance:*

Unbiasedness:

$$\mathbb{E}_{B_k^t \subseteq D_k} [g_{w,k}^t] = \nabla \mathcal{L}_k^t(w_k^t). \quad (9)$$

Bounded variance:

$$\mathbb{E}_{B_k^t \subseteq D_k} [\|\nabla \mathcal{L}_k^t(w_k^t; B_k^t) - \nabla \mathcal{L}_k^t(w_k^t)\|^2] \leq \sigma^2 \quad (10)$$

**Assumption 3. Bounded Gradient in Global Alignment.** *Inspired by the theoretical bound on semantic alignment loss variation per round in FedProto, the first-stage local update iterations in FedKWAZ are defined, during which global semantic and decision alignment is performed on each client's local model. The resulting gradient shift from each single round is uniformly upper-bounded as:*

$$\|\nabla \mathcal{L}_{k, \text{GlobalAlign}}^{tE+e}\|^2 \leq \delta^2, \forall e \in \{0, 1, \dots, E_A - 1\}, k \in \{0, 1, \dots, N - 1\}. \quad (11)$$

**Assumption 4. Bounded Gradient in KWAZ Alignment.** *Similarly, in the second stage of FedKWAZ (the KWAZ learning stage), the knowledge alignment operations (including SWAZ and DWAZ) performed between the local heterogeneous model and the proxy model on the client side are assumed to exhibit a unified bound on per-step gradient variation:*

$$\|\nabla \mathcal{L}_{k, \text{KWAZAlign}}^{tE+e}\|^2 \leq \gamma^2, \forall e \in \{E_A, E_A + 1, \dots, E_A + E_B - 1\}, k \in \{0, 1, \dots, N - 1\} \quad (12)$$

**Lemma 1. Stage-I Bias.** *Given Assumptions 1, 2 and 3, After the completion of the first stage, the following inequality is satisfied for any client:*

$$\mathbb{E}[\mathcal{L}_k(w_k^{tE+E_A})] \leq \mathcal{L}_k(w_k^{tE+0}) - \left(\eta - \frac{L_1 \eta^2}{2}\right) \sum_{e=0}^{E_A-1} \|\nabla \mathcal{L}_k(w_k^{tE+e})\|_2^2 + \frac{L_1 E_A \eta^2 \sigma^2}{2} + \eta E_A \delta^2 \quad (13)$$

**Lemma 2. Stage-II Bias.** *Given Assumptions 1, 2 and 4, After the completion of the second stage, the following inequality is satisfied for any client:*

$$\begin{aligned} \mathbb{E}[\mathcal{L}_k(w_k^{tE+E_A+E_B})] &\leq \mathbb{E}[\mathcal{L}_k(w_k^{tE+E_A})] - \left(\eta - \frac{L_1 \eta^2}{2}\right) \sum_{e=E_A}^{E_A+E_B-1} \|\nabla \mathcal{L}_k(w_k^{tE+e})\|_2^2 \\ &\quad + \frac{L_1 E_B \eta^2 \sigma^2}{2} + \eta E_B \gamma^2 \end{aligned} \quad (14)$$



**Theorem 1. One Complete Round of FL.** Given the above lemma, for any client, after the two-stage mutual learning has been completed, the following inequality holds:

$$\begin{aligned} \mathbb{E}[\mathcal{L}_k(w_k^{tE+E_A+E_B})] &\leq \mathcal{L}_k(w_k^{tE+0}) - \left(\eta - \frac{L_1\eta^2}{2}\right) \sum_{e=0}^{E_A+E_B-1} \|\nabla \mathcal{L}_k(w_k^{tE+e})\|_2^2 \\ &\quad + \frac{L_1(E_A + E_B)\eta^2\sigma^2}{2} + \eta(E_A\delta^2 + E_B\gamma^2) \end{aligned} \quad (15)$$

**Theorem 2. Non-convex Convergence Rate of FedKWAZ.** Given Theorem 1, For any client and any constant  $\epsilon > 0$ , the following inequality holds:

$$\begin{aligned} &\frac{1}{T} \sum_{t=0}^{T-1} \sum_{e=0}^{E_A+E_B-1} \|\nabla \mathcal{L}_k(w_k^{tE+e})\|_2^2 \\ &\leq \frac{\frac{1}{T} \sum_{t=0}^{T-1} \left[ \mathcal{L}_k(w_k^{tE+0}) - \mathbb{E}[\mathcal{L}_k(w_k^{tE+E_A+E_B})] \right] + \frac{L_1(E_A+E_B)\eta^2\sigma^2}{2} + \eta(E_A\delta^2 + E_B\gamma^2)}{\eta - \frac{L_1\eta^2}{2}} < \epsilon \\ &\text{s.t. } 0 < \eta < \frac{2(\epsilon - (E_A\delta^2 + E_B\gamma^2))}{L_1(\epsilon + (E_A + E_B)\sigma^2)}. \end{aligned} \quad (16)$$

### C.1 Proof of Lemma 1

*Proof.* In the  $e$ -th local update of the first stage ( $e \in 0, 1, \dots, E_A - 1$ ), any client  $k$  is considered, and the result can be derived based on the Lipschitz smoothness in 1.

$$\begin{aligned} \mathbb{E}[\mathcal{L}_k(w_k^{tE+e+1})] &\stackrel{(a)}{\leq} \mathbb{E}[\mathcal{L}_k(w_k^{tE+e}) + \langle \nabla \mathcal{L}_k(w_k^{tE+e}), w_k^{tE+e+1} - w_k^{tE+e} \rangle + \frac{L_1}{2} \|w_k^{tE+e+1} - w_k^{tE+e}\|_2^2] \\ &\stackrel{(b)}{=} \mathcal{L}_k(w_k^{tE+e}) + \mathbb{E}[\langle \nabla \mathcal{L}_k(w_k^{tE+e}), -\eta g_k^{tE+e} \rangle + \frac{L_1}{2} \|\eta g_k^{tE+e}\|_2^2] \\ &= \mathcal{L}_k(w_k^{tE+e}) - \eta \mathbb{E}[\langle \nabla \mathcal{L}_k(w_k^{tE+e}), g_k^{tE+e} \rangle] + \frac{L_1\eta^2}{2} \mathbb{E}[\|g_k^{tE+e}\|_2^2] \\ &\stackrel{(c)}{=} \mathcal{L}_k(w_k^{tE+e}) - \eta \|\nabla \mathcal{L}_k(w_k^{tE+e})\|_2^2 + \frac{L_1\eta^2}{2} \mathbb{E}[\|g_k^{tE+e}\|_2^2] \\ &\stackrel{(d)}{\leq} \mathcal{L}_k(w_k^{tE+e}) - \eta \|\nabla \mathcal{L}_k(w_k^{tE+e})\|_2^2 + \frac{L_1\eta^2}{2} (\|\nabla \mathcal{L}_k(w_k^{tE+e})\|_2^2 + \sigma^2) \\ &= \mathcal{L}_k(w_k^{tE+e}) - \left(\eta - \frac{L_1\eta^2}{2}\right) \|\nabla \mathcal{L}_k(w_k^{tE+e})\|_2^2 + \frac{L_1\eta^2\sigma^2}{2} \end{aligned}$$

Step (a): Based on Assumption 1,  $\mathcal{L}_k$  is approximated near  $w_k^{tE+e}$  using a first-order Taylor expansion.

Step (b): The update rule  $w_k^{tE+e+1} = w_k^{tE+e} - \eta g_k^{tE+e}$  is applied, where  $g_k^{tE+e}$  denotes the stochastic gradient over the current batch.

Step (c): Based on Assumption 2 (unbiased gradients).

Step (d): From Assumption 2 (bounded variance of the gradient), it holds that

$$\mathbb{E}[\|g_k^{tE+e} - \nabla \mathcal{L}_k(w_k^{tE+e})\|_2^2] \leq \sigma^2 \quad (17)$$

From this, the following upper bound is derived:

$$\mathbb{E}[\|g_k^{tE+e}\|_2^2] \leq \|\nabla \mathcal{L}_k(w_k^{tE+e})\|_2^2 + \sigma^2 \quad (18)$$

By summing the single-step inequalities from  $e = 0$  to  $e = E_A - 1$ , the following result is obtained:

$$\mathbb{E}[\mathcal{L}_k(w_k^{tE+E_A})] \leq \mathcal{L}_k(w_k^{tE+0}) - \left(\eta - \frac{L_1\eta^2}{2}\right) \sum_{e=0}^{E_A-1} \|\nabla \mathcal{L}_k(w_k^{tE+e})\|_2^2 + \frac{L_1E_A\eta^2\sigma^2}{2} \quad (19)$$

Furthermore, based on Assumption 3 (bounded global alignment gradient in the first stage), a per-round gradient variation upper bound  $\delta^2$  is introduced, leading to an extra term  $\eta E_A \delta^2$  in the total sum. Finally, Lemma 1 is fully expressed as Eq. 13.  $\square$

## C.2 Proof of Lemma 2

*Proof.* Similarly, based on Assumptions 1, 2, and 4, after the second stage of FedKWAZ (KWAZ knowledge alignment stage), the expected local loss at any client  $k$  is shown to satisfy Eq. 14.  $\square$

## C.3 Proof of Theorem 1

*Proof.* By substituting Lemma 1 into the right-hand side of the inequality in Lemma 2, Eq. 15 can be obtained.  $\square$

## C.4 Proof of Theorem 2

*Proof.* The left and right sides of Theorem 1 are swapped, and the gradient terms are rearranged:

$$\begin{aligned} & \sum_{e=0}^{E_A+E_B-1} \|\nabla \mathcal{L}_k(w_k^{tE+e})\|_2^2 \\ & \leq \frac{\mathcal{L}_k(w_k^{tE+0}) - \mathbb{E}[\mathcal{L}_k(w_k^{tE+E_A+E_B})] + \frac{L_1(E_A+E_B)\eta^2\sigma^2}{2} + \eta(E_A\delta^2 + E_B\gamma^2)}{\eta - \frac{L_1\eta^2}{2}}. \end{aligned} \quad (20)$$

In the total  $T$  rounds of federated communication training, the expectation of the above inequality is taken for  $t$  from 0 to  $T-1$  and summed, yielding:

$$\begin{aligned} & \frac{1}{T} \sum_{i=0}^{T-1} \sum_{e=0}^{E_A+E_B-1} \|\nabla \mathcal{L}_k(w_k^{tE+e})\|_2^2 \\ & \leq \frac{\frac{1}{T} \sum_{t=0}^{T-1} \left[ \mathcal{L}_k(w_k^{tE+0}) - \mathbb{E}[\mathcal{L}_k(w_k^{tE+E_A+E_B})] \right] + \frac{L_1(E_A+E_B)\eta^2\sigma^2}{2} + \eta(E_A\delta^2 + E_B\gamma^2)}{\eta - \frac{L_1\eta^2}{2}}. \end{aligned} \quad (21)$$

The difference between the loss at the initial time and the optimal loss is defined as:

$$\Delta = \mathcal{L}_0 - \mathcal{L}^* > 0 \quad (22)$$

The expected loss function over  $T$  rounds is expressed as:

$$\frac{1}{T} \sum_{t=0}^{T-1} \left[ \mathcal{L}_k(w_k^{tE+0}) - \mathbb{E}[\mathcal{L}_k(w_k^{tE+E_A+E_B})] \right] \leq \frac{\Delta}{T} \quad (23)$$

Hence, the original inequality can be further simplified as:

$$\frac{1}{T} \sum_{t=0}^{T-1} \sum_{e=0}^{E_A+E_B-1} \|\nabla \mathcal{L}_k(w_k^{tE+e})\|_2^2 \leq \frac{\frac{\Delta}{T} + \frac{L_1(E_A+E_B)\eta^2\sigma^2}{2} + \eta(E_A\delta^2 + E_B\gamma^2)}{\eta - \frac{L_1\eta^2}{2}}. \quad (24)$$

Let the expected norm of the modulus in the above equation be expected to converge to a constant  $\epsilon$ :

$$\frac{\frac{\Delta}{T} + \frac{L_1(E_A+E_B)\eta^2\sigma^2}{2} + \eta(E_A\delta^2 + E_B\gamma^2)}{\eta - \frac{L_1\eta^2}{2}} < \epsilon \quad (25)$$

Since the number of training iterations  $T > 0$ , and  $\Delta > 0$ , the denominator must satisfy:

$$\epsilon\left(\eta - \frac{L_1\eta^2}{2}\right) - \frac{L_1(E_A + E_B)\eta^2\sigma^2}{2} - \eta(E_A\delta^2 + E_B\gamma^2) > 0 \quad (26)$$

Thus, the upper bound of  $\eta$  is obtained as:

$$0 < \eta < \frac{2(\epsilon - (E_A\delta^2 + E_B\gamma^2))}{L_1(\epsilon + (E_A + E_B)\sigma^2)} \quad (27)$$

$\square$

Since all of the quantities  $\epsilon$ ,  $L_1$ ,  $\sigma^2$ ,  $\delta^2$ , and  $\gamma^2$  are positive finite constants, the constraint on the learning rate  $\eta$  exists in a non-empty solution set. When the learning rate  $\eta$  satisfies the above conditions, the expected norm of the local loss for any client can converge to the constant  $\epsilon$ . According to the right-hand side of the above equation, except for the term divided by  $\frac{1}{T}$ , the remaining terms are constants. Therefore, the non-convex convergence rate of FedKWAZ is achieved as:  $O(1/T)$ .

## D Additional Experimental Details

### D.1 Experimental Environment

All experiments are conducted on the PyTorch platform. The experiments are executed on four NVIDIA GeForce 4090 GPUs (24GB memory) across five supervised image classification datasets <sup>1</sup>.

### D.2 Datasets

The sources of the datasets are detailed. The experiments are conducted based on five public multi-class datasets, covering natural and medical image recognition tasks, including:

Cifar10 (<https://pytorch.org/vision/main/generated/torchvision.datasets.CIFAR10.html>),  
Cifar100 (<https://pytorch.org/vision/stable/generated/torchvision.datasets.CIFAR100.html>),  
Flowers102 (<https://pytorch.org/vision/stable/generated/torchvision.datasets.Flowers102.html>),  
Tiny-ImageNet (<http://cs231n.stanford.edu/tiny-imagenet-200.zip>),  
and Skin-Lesions-14 (<https://www.kaggle.com/datasets/ahmedxc4/skin-ds>) are utilized.

### D.3 Hyperparameter Settings

In addition to the hyperparameter settings provided in the main text, the hyperparameter configurations for each baseline method are also followed according to their original publications. Specifically, LG-FedAvg is configured with no additional hyperparameters; for FedGen, the noise dimension is set to 32, the generator learning rate is set to 0.1, and the hidden dimension is aligned with the feature dimension  $K$ , with 100 training rounds on the server. The distillation hyperparameters in FML are set as  $\alpha = 0.5$  and  $\beta = 0.5$ ; in FedKD, the proxy model’s learning rate is set to 0.01 to match the client; the temperature range for distillation is set to  $T_{\text{start}} = 0.95$  and  $T_{\text{end}} = 0.95$ ; for FedDistill,  $\gamma = 1$  is applied. FedProto is configured with  $\lambda = 0.1$ ; in FedGH, the server learning rate is set to 0.01 to match the client; the representation dimension of the proxy model in FedMRL is set to 256. For FedTGP,  $\lambda$  is set to 0.1, the distillation margin threshold  $\tau$  to 100, and the server training rounds to 100; in FedKTL,  $K$  is set to  $C$ ,  $\mu = 50$ ,  $\lambda = 1$ , with server learning rate  $\eta_s = 0.01$ , batch size  $B_s = 100$ , and server training rounds  $E_s = 100$ . Except for FedGen and FedKTL, where server-side training is performed using the Adam [2] optimizer, SGD [10] is applied for client- and server-side training in all other methods. In FedKWAZ, the HAPM module is employed to automatically search for optimal mixing strength parameters ( $\alpha$  or  $\beta$ ) using KDP within the range 0.1, 0.5, 1.0. The spatial granularity  $G$  is used to divide each input image into  $\sqrt{G} \times \sqrt{G}$  local patches. Specifically, for CIFAR10 and CIFAR100 (input size  $3 \times 32 \times 32$ ), as well as Flowers102 and Tiny-ImageNet (input size  $3 \times 64 \times 64$ ),  $G \in 64, 16, 4$ ; for Skin-Lesions-14 (input size  $3 \times 28 \times 28$ ),  $G \in 49, 16, 4$ . In FedKWAZ, the KWAZ update frequency (i.e., the interval  $k$  for KDP-based HAPM parameter search) is fixed to once every 30 rounds to balance the ability to absorb prior knowledge and explore novel knowledge; the distillation temperature  $\tau$  is set to 4. The training epoch  $E_A$  and  $E_B$  for the first and second stages are uniformly set to 1.

### D.4 Detailed Setting of Model Heterogeneity

The primary test configuration for feature extractor heterogeneity is defined as HtFE<sub>8</sub>, consisting of eight representative network architectures: 4-layer CNN [4], GoogleNet [6], MobileNet\_v2 [5], ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152 [1]. Networks of ResNet18 and deeper are classified under the ResNet series and are used to construct multi-level heterogeneous configurations, including: HtFE<sub>1</sub> (containing only ResNet4, used to simulate a fully homogeneous scenario), HtFE<sub>2</sub> (composed of CNN and ResNet18, indicating slight heterogeneity), HtFE<sub>4</sub> (integrating CNN,

<sup>1</sup>Code is available at: <https://github.com/ysml666/FedKWAZ>

GoogleNet, MobileNet\_v2, and ResNet18 to simulate moderate heterogeneity), and HtFE<sub>9</sub> (configured as extremely heterogeneous by combining ResNet4, ResNet6, ResNet8, ResNet10, ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152).

Regarding classifier heterogeneity, the HtC<sub>4</sub> scenario is constructed to include four types of classifiers composed only of fully connected (FC) layers, to simulate structural heterogeneity in decision-making. The four architectures are defined as: (1) single-layer FC (100-d), (2) two-layer FC (512-d → 100-d) with a 512-d hidden layer, (3) two-layer FC (256-d → 100-d), and (4) two-layer FC (128-d → 100-d). In the HtFE<sub>8</sub>-HtC<sub>4</sub> scenario, the eight types of feature extractors and four types of classifiers are cross-combined according to client indices, thereby forming a test environment with simultaneous heterogeneity in both feature extraction and decision modules.

In mutual learning schemes, the proxy model structure is defined as a simple 4-layer CNN to suit methods such as FedKD, FML, and FedMRL that periodically upload parameters of the proxy models to the server, aiming to reduce communication overhead. In contrast, FedKWAZ avoids proxy model transmission entirely and instead constructs global semantic prototypes and decision anchors by aggregating local class-wise features and logits, the proxy model is still uniformly set as a 4-layer CNN to ensure consistency and fairness in experimental comparisons, and to eliminate the influence of model complexity as a confounding variable.

## E Additional Experimental Results

### E.1 Impact of Feature Dimensions

As shown in Table 1, most algorithms exhibit accuracy gains as the feature dimension increases from  $K = 64$  to  $K = 256$ . At  $K = 256$ , FedKWAZ attains an accuracy of 50.95%, surpassing FedKTL by 5.16% (Figure 1), indicating its strong capability in leveraging high-dimensional representations for effective knowledge interaction. When the feature dimension is further increased to  $K = 1024$ , performance declines are observed in several methods, which can be attributed to elevated sparsity and reduced discriminative efficiency in overly expanded feature spaces. In contrast, FedKWAZ maintains competitive performance with 50.87% accuracy, supported by its KWAZ-guided knowledge coordination.

In addition, several federated learning baselines are outperformed by the Local method under certain configurations, as negative transfer under dual heterogeneity in data and model structures compromises collaborative training. FedKWAZ, by incorporating KWAZ-guided modeling of representation and decision discrepancies, effectively alleviates cross-model knowledge transfer bottlenecks and maintains consistent performance across diverse tasks and configurations.

Table 1: Impact of feature dimensions ( $K$ ) on HtFE<sub>8</sub> model group performance on Cifar100.

|            | $K = 64$          | $K = 256$         | $K = 1024$        |
|------------|-------------------|-------------------|-------------------|
| Local      | 39.24±0.14        | 40.92±0.11        | 40.25±0.18        |
| FedDistill | 38.93±0.20        | 44.10±0.15        | 42.56±0.23        |
| LG-FedAvg  | 39.66±0.22        | 40.15±0.14        | 41.25±0.19        |
| FedGen     | 38.75±0.15        | 40.23±0.19        | 40.43±0.11        |
| FedKD      | 40.54±0.16        | 40.26±0.17        | 41.08±0.10        |
| FedProto   | 31.84±0.10        | 35.63±0.12        | 34.14±0.14        |
| FML        | 38.40±0.18        | 40.80±0.11        | 40.59±0.19        |
| FedGH      | 38.19±0.18        | 40.01±0.15        | 38.48±0.17        |
| FedMRL     | 39.06±0.14        | 41.90±0.16        | 42.95±0.12        |
| FedTGP     | 47.05±0.17        | 47.87±0.27        | 47.43±0.24        |
| FedKTL     | 45.98±0.15        | 45.79±0.14        | 46.76±0.17        |
| FedKWAZ    | <b>50.04±0.13</b> | <b>50.95±0.16</b> | <b>50.87±0.12</b> |

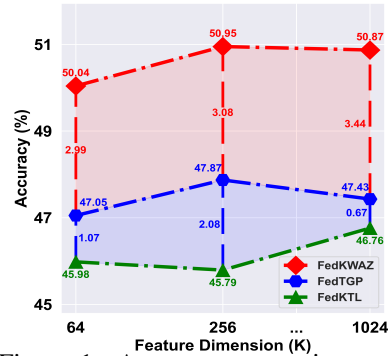


Figure 1: Accuracy comparison among FedKWAZ, FedTGP, and FedKTL.

### E.2 Homogeneous Models Setting

To further investigate the role of data heterogeneity in isolation, all clients are configured with identical model architectures across three homogeneous settings: ResNet10, ResNet18, and ResNet34, in

addition to the original  $\text{HtFE}_1$  (ResNet4) baseline listed in Table ?? . By removing model-level heterogeneity, these experiments focus solely on the impact of non-IID data distributions on federated knowledge transfer. As shown in Table 2, FedKWAZ consistently outperforms all baseline methods under each homogeneous configuration. These results demonstrate that, even without structural differences across models, the proposed dual-stage mutual learning framework remains effective by explicitly aligning both representation semantics and decision behaviors across clients. This joint alignment enables FedKWAZ to bridge knowledge gaps induced purely by data distribution shifts, thereby ensuring stable performance improvements under non-IID conditions.

### E.3 Performance on Fashion-MNIST under Pathological and Dirichlet Data Settings

On the Fashion-MNIST [9] dataset, the  $\text{HtCNN}_8$  model group is adopted to accommodate the grayscale single-channel input, with partition details listed in Table 4. A comprehensive evaluation of all methods is conducted under both Pathological and Practical non-IID settings. As shown in Table 3, FedKWAZ consistently yields the highest accuracy in both scenarios. Although FMNIST presents relatively modest classification complexity and most methods perform well, FedKWAZ maintains a consistent performance lead. In addition, Figure 2 presents the t-SNE [7] visualization of learned feature representations under the pathological setting for representative mutual learning baselines (FML, FedKD, FedMRL) and FedKWAZ. Compared to these baselines, the features produced by FedKWAZ exhibit stronger intra-class compactness and inter-class separability, reflecting enhanced semantic consistency under model heterogeneity and data distribution shift, and supporting improved cross-model knowledge transfer.

Table 2: Performance using homogeneous models on Cifar100 in the practical setting.

| Architectures | ResNet10          | ResNet18          | ResNet34          |
|---------------|-------------------|-------------------|-------------------|
| Local         | 45.38±0.16        | 42.57±0.12        | 41.62±0.14        |
| FedDistill    | 44.78±0.18        | 44.12±0.21        | 43.62±0.23        |
| LG-FedAvg     | 47.11±0.17        | 44.53±0.15        | 44.04±0.20        |
| FedGen        | 47.02±0.22        | 44.53±0.13        | 44.27±0.11        |
| FedKD         | 45.13±0.11        | 41.32±0.15        | 40.26±0.12        |
| FedProto      | 40.67±0.25        | 40.23±0.18        | 38.02±0.15        |
| FML           | 46.37±0.16        | 43.07±0.17        | 40.25±0.12        |
| FedGH         | 45.30±0.12        | 43.29±0.14        | 41.84±0.10        |
| FedMRL        | 47.36±0.09        | 45.67±0.10        | 45.40±0.12        |
| FedTGP        | 47.05±0.38        | 45.79±0.35        | 47.43±0.33        |
| FedKTL        | 51.20±0.15        | 50.10±0.14        | 48.17±0.12        |
| FedKWAZ       | <b>52.35±0.11</b> | <b>52.47±0.14</b> | <b>51.75±0.12</b> |

Table 3: Performance on FMNIST using the  $\text{HtCNN}_8$ .

| Settings   | Pathological Setting | Practical Setting |
|------------|----------------------|-------------------|
| Local      | 99.38±0.05           | 97.22±0.09        |
| FedDistill | 99.42±0.04           | <b>97.44±0.03</b> |
| LG-FedAvg  | 99.37±0.05           | 97.22±0.04        |
| FedGen     | 99.34±0.06           | 97.34±0.05        |
| FedKD      | 99.40±0.07           | 97.36±0.03        |
| FedProto   | 99.39±0.03           | 97.35±0.02        |
| FML        | 99.41±0.05           | 97.34±0.04        |
| FedGH      | 99.38±0.04           | 97.36±0.03        |
| FedMRL     | <b>99.45±0.05</b>    | 97.08±0.04        |
| FedTGP     | <b>99.52±0.06</b>    | <b>97.53±0.05</b> |
| FedKWAZ    | <b>99.60±0.03</b>    | <b>97.61±0.04</b> |

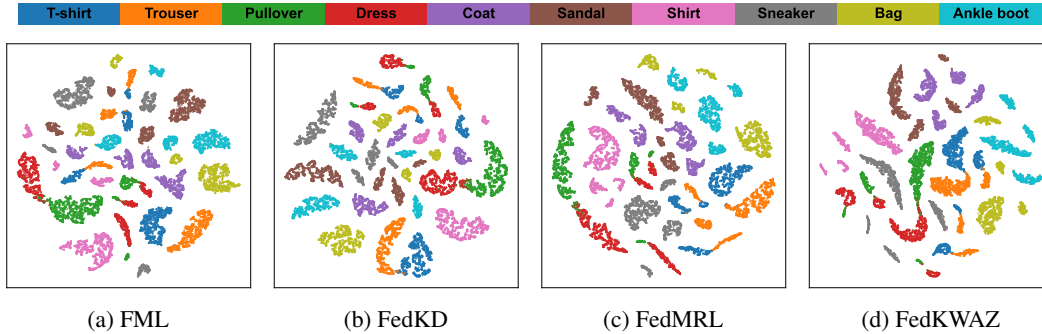


Figure 2: T-SNE visualization of features extracted by FML, FedKD, FedMRL and FedKWAZ on FMNIST under pathological partition.

### E.4 Performance under Low Client Participation Rates and High Client Drop Rates

On the Cifar10 dataset (Dirichlet distribution, 100 clients), client participation is systematically restricted to simulate federated scenarios with limited and unstable communication. Participation rates of 5% and 10% are adopted, corresponding to only 5 or 10 randomly selected clients contributing to each round, while aggregation uses updates only from the clients participating in that round. For

Table 4: The model architectures in the HtCNN<sub>8</sub> group. Convolutional layers are represented as “[5 × 5, 32]” indicating a convolution with kernel size 5 × 5 and 32 output channels, and “2 × 2 max pooling” refers to a max pooling layer with kernel size 2 × 2.

|      | Sequentially Connected Feature Extractors   | Classifiers |
|------|---|-------------|
| CNN1 | Conv (5 × 5, 32), 2 × 2 max pool, 512-d fc  | 10-d fc     |
| CNN2 | Conv (5 × 5, 32), 2 × 2 max pool, Conv (5 × 5, 64), 2 × 2 max pool, 512-d fc                | 10-d fc     |
| CNN3 | Conv (5 × 5, 32), 2 × 2 max pool, 2 × 512-d fc  | 10-d fc     |
| CNN4 | Conv (5 × 5, 32), 2 × 2 max pool, Conv (5 × 5, 64), 2 × 2 max pool, 2 × 512-d fc            | 10-d fc     |
| CNN5 | Conv (5 × 5, 32), 2 × 2 max pool, 1024-d fc, 512-d fc                                       | 10-d fc     |
| CNN6 | Conv (5 × 5, 32), 2 × 2 max pool, Conv (5 × 5, 64), 2 × 2 max pool, 1024-d fc, 512-d fc     | 10-d fc     |
| CNN7 | Conv (5 × 5, 32), 2 × 2 max pool, 1024-d fc × 2, 512-d fc                                   | 10-d fc     |
| CNN8 | Conv (5 × 5, 32), 2 × 2 max pool, Conv (5 × 5, 64), 2 × 2 max pool, 1024-d fc, 512-d fc × 2 | 10-d fc     |

FML, FedKD, and FedMRL, global proxy models are constructed by aggregating local proxy models from all clients. In contrast, FedKWAZ performs global aggregation using class-wise prototypes and logits from all clients to form semantic representations and decision anchors. As reported in Table 5, overall performance degrades as participation rate decreases. Nonetheless, FedKWAZ consistently achieves the highest accuracy at both 5% and 10% settings, demonstrating strong resilience to client sparsity. Figure 3 further illustrates the prediction matrix at 10% participation, where FedKWAZ exhibits sharper diagonal confidence compared to other methods, indicating improved inter-class separation and stronger decision reliability under sparse client engagement.

Table 5: Performance of FML, FedKD, FedMRL and FedKWAZ under low client participation rates and high client drop rates.

|         | Client Participation Rate |                   | Client Drop Rate  |                   |
|---------|---------------------------|-------------------|-------------------|-------------------|
|         | 5%                        | 10%               | 90%               | 95%               |
| FML     | 79.54±0.10                | 80.40±0.12        | 81.55±0.14        | 81.12±0.11        |
| FedKD   | 79.41±0.11                | 79.67±0.10        | 80.66±0.16        | 80.48±0.13        |
| FedMRL  | 80.92±0.12                | 81.08±0.15        | 81.65±0.13        | 81.39±0.10        |
| FedKWAZ | <b>82.85±0.09</b>         | <b>83.71±0.11</b> | <b>84.37±0.12</b> | <b>83.93±0.13</b> |

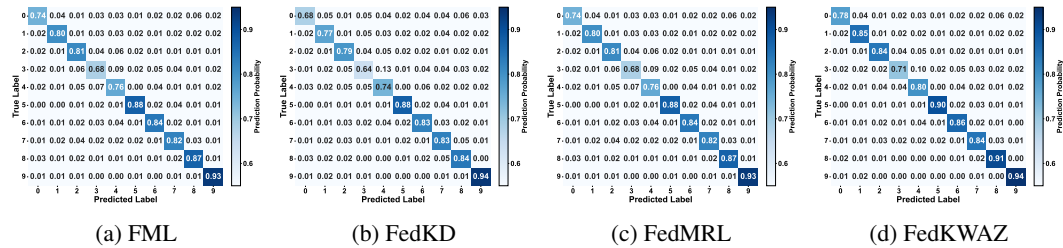


Figure 3: Confusion matrices of FML, FedKD, FedMRL, and FedKWAZ on Cifar10 under dirichlet partition with a client participation rate of 0.1, showing the prediction probability for each class.

To further evaluate communication disruptions, a Client Drop Rate experiment is conducted. In this setting, all 100 clients participate in local training, but only a portion are able to upload their knowledge due to simulated dropout. Dropout rates of 90% and 95% are used, meaning that only 10 or 5 clients successfully contribute to global aggregation. As shown in Table 5, these settings yield better accuracy than the 10% and 5% participation scenarios, suggesting that consistent local participation—even under partial upload failure—helps preserve training efficacy and partially compensates for reduced aggregation scale.

These results collectively confirm that FedKWAZ, empowered by its dual-stage knowledge transfer mechanism and lightweight semantic anchoring, maintains robust knowledge integration even under extreme communication sparsity and unstable client availability, effectively addressing practical challenges in real-world federated deployments.

### E.5 Impact of SWAZ and DWAZ

The individual contributions of SWAZ and DWAZ to the second-stage mutual learning in FedKWAZ are further investigated. Specifically, experiments are conducted by selectively enabling either SWAZ or DWAZ while disabling the other, and performance is assessed on the Skin-Lesions-14 and Flowers102 datasets under both Dirichlet and Pathological data partitions. As reported in Figure 4, DWAZ consistently achieves higher accuracy than SWAZ, indicating that decision-level alignment exerts a more immediate influence on cross-model knowledge transfer. This observation highlights the critical role of decision space consistency in federated mutual learning. Furthermore, the full FedKWAZ—combining both SWAZ and DWAZ—achieves the highest performance across all scenarios, confirming the complementary nature of semantic and decision-level discrepancy modeling. These findings underscore the necessity of jointly capturing both representational and predictive misalignments to optimize knowledge transfer between heterogeneous models.

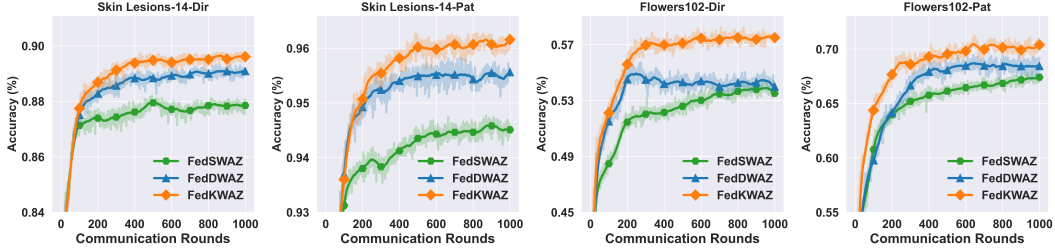


Figure 4: The contributions of SWAZ and DWAZ to performance.

### E.6 Impact of Feature Shift

To further assess the robustness of each method under feature shift scenarios, a heterogeneous data experiment is constructed in which each client is provided with a complete set of class labels, but the visual characteristics of the data—such as style, texture, and viewpoint—differ substantially across domains. Two widely adopted domain generalization benchmarks are selected: PACS [3], comprising 9,991 images from 7 categories across 4 domains (Art, Cartoon, Photo, Sketch), and OfficeHome [8], containing 15,588 images from 65 categories over 4 domains (Art, Clipart, Product, Real). For each dataset, samples are split into training and test sets with a 3:1 ratio, and each domain is treated as a distinct client, simulating federated feature heterogeneity.

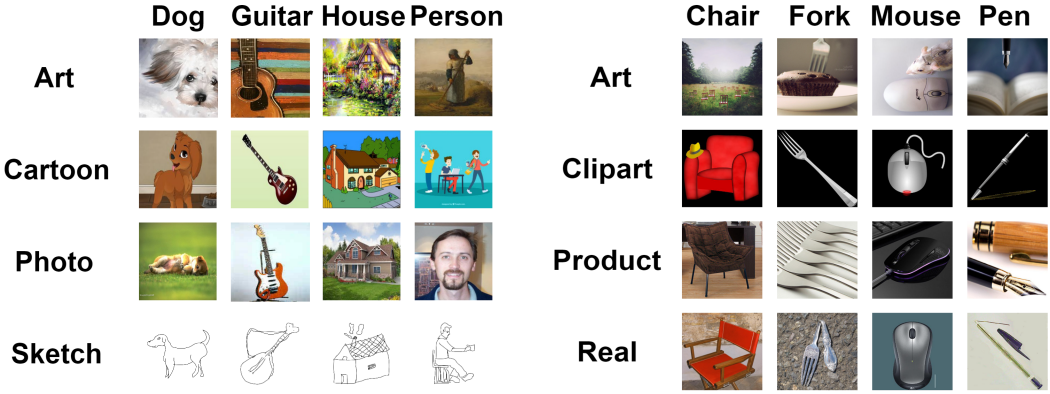
In model allocation, the HtFE<sub>4</sub> architecture configuration is adopted, where heterogeneous backbones—including a 4-layer CNN, GoogleNet, MobileNet\_v2, and ResNet18—are deployed across clients to emulate the hardware and model diversity typical in real-world federated environments. Figure 5 illustrates the domain-induced feature shift across different domains in PACS and OfficeHome for a representative category, highlighting the substantial variation in visual statistics. Table 6 summarizes the per-domain test accuracy at the final communication round and the overall average accuracy achieved at the optimal round for each method. Notably, the average accuracy is not obtained via direct averaging over individual client scores but is instead computed based on the total number of correct predictions across all test samples, yielding a more comprehensive and balanced measure of global performance.

As presented in Table 6 and Figure 6, FedKWAZ consistently demonstrates faster convergence, improved communication efficiency, and higher per-domain accuracies. In addition, it achieves the highest aggregated performance, significantly surpassing all baselines. These results suggest that FedKWAZ effectively addresses the representational and decision-level mismatches induced by feature shift through its dynamic KWAZ-based localization and targeted enhancement strategy, thereby enabling more reliable knowledge transfer across heterogeneous clients.



Table 6: Test accuracy (%) comparison of various federated learning methods on PACS and OfficeHome datasets. Each column represents the accuracy on one domain, while "Avg" indicates the mean accuracy across all domains.

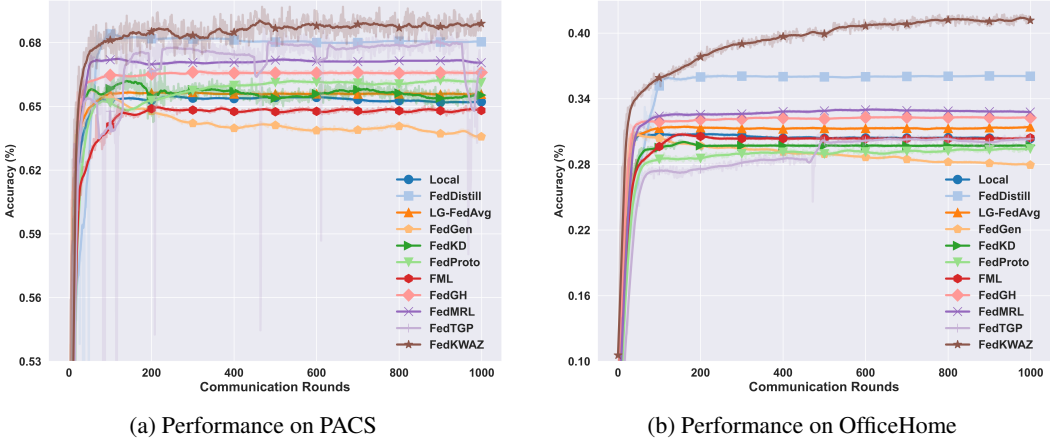
| Methods    | PACS       |            |            |            |            | OfficeHome |            |            |            |            |
|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
|            | Art        | Cartoon    | Photo      | Sketch     | Avg        | Art        | Clipart    | Product    | Real       | Avg        |
| Local      | 40.23±0.09 | 78.64±0.11 | 58.61±0.07 | 73.04±0.11 | 65.51±0.18 | 14.50±0.14 | 47.25±0.14 | 33.96±0.08 | 18.53±0.17 | 30.96±0.18 |
| FedDistill | 42.66±0.20 | 78.60±0.14 | 60.77±0.21 | 77.72±0.25 | 68.63±0.24 | 16.14±0.15 | 49.73±0.20 | 48.47±0.13 | 20.83±0.15 | 36.16±0.16 |
| LG-FedAvg  | 40.04±0.14 | 76.62±0.23 | 60.77±0.23 | 74.47±0.19 | 65.79±0.15 | 14.00±0.19 | 46.43±0.13 | 37.84±0.10 | 19.08±0.12 | 31.57±0.08 |
| FedGen     | 40.43±0.18 | 78.84±0.31 | 57.18±0.08 | 69.38±0.16 | 65.99±0.17 | 12.85±0.21 | 47.25±0.21 | 33.96±0.30 | 11.10±0.24 | 31.08±0.27 |
| FedKD      | 42.40±0.08 | 78.50±0.21 | 62.44±0.23 | 76.16±0.15 | 67.03±0.16 | 16.31±0.18 | 45.60±0.08 | 36.76±0.16 | 18.62±0.17 | 30.18±0.15 |
| FedProto   | 41.75±0.21 | 77.65±0.20 | 61.57±0.24 | 73.75±0.26 | 66.31±0.24 | 16.39±0.08 | 48.64±0.21 | 33.06±0.23 | 14.59±0.15 | 29.67±0.18 |
| FML        | 42.21±0.18 | 78.67±0.08 | 59.57±0.10 | 71.92±0.22 | 65.11±0.15 | 15.17±0.13 | 43.95±0.26 | 33.34±0.19 | 17.42±0.23 | 30.83±0.14 |
| FedGH      | 39.65±0.15 | 78.52±0.13 | 59.33±0.26 | 76.50±0.19 | 66.67±0.23 | 15.49±0.15 | 47.53±0.16 | 41.89±0.18 | 16.42±0.08 | 32.39±0.10 |
| FedMRL     | 39.45±0.17 | 78.40±0.15 | 60.29±0.13 | 76.40±0.26 | 67.39±0.19 | 15.82±0.23 | 51.77±0.14 | 37.59±0.13 | 17.81±0.26 | 33.11±0.19 |
| FedTGP     | 38.47±0.23 | 76.45±0.34 | 63.16±0.30 | 78.94±0.32 | 68.71±0.36 | 14.28±0.22 | 47.60±0.28 | 33.77±0.34 | 18.26±0.26 | 30.73±0.32 |
| FedKWAZ    | 42.77±0.21 | 78.91±0.13 | 63.51±0.26 | 79.04±0.19 | 69.43±0.23 | 17.16±0.15 | 54.96±0.48 | 55.50±0.18 | 28.90±0.08 | 41.70±0.10 |



(a) Domain shift in PACS across four domains.

(b) Domain shift in OfficeHome across four domains.

Figure 5: Visualization of feature shift across different domains in PACS and OfficeHome datasets.



(a) Performance on PACS

(b) Performance on OfficeHome

Figure 6: Comparison of test accuracy across various methods on PACS and OfficeHome datasets.

## E.7 Effectiveness of HAPM vs. Hard Sample Selection

To further verify the effectiveness of the proposed Hierarchical Adaptive Patch Mixing (HAPM) module, we conducted additional comparative experiments to evaluate its advantage over naïve image distillation and local hard-sample selection strategies in facilitating cross-model knowledge transfer. The experiments were performed on two representative datasets, CIFAR-10 and Flowers102.

The compared methods include:



FedKD (Original): The private and proxy models align their features and logits using only original local images.

FedKD + Local Hard Sample Enhancement: The top 10%, 30%, and 50% local samples with the highest feature MSE and logits KL divergence were selected for enhanced distillation.

FedKWAZ (Full Framework): HAPM generates perturbed mixed samples, and KDP identifies semantic–decision weak zones (SWAZ and DWAZ) for fine-grained mutual learning.

Table 7: Performance comparison between FedKD, hard sample-enhanced FedKD (top 10%, 30%, 50% samples), and FedKWAZ on CIFAR-10 and Flowers102 datasets.

| Model                       | CIFAR-10 | Flowers102 |
|-----------------------------|----------|------------|
| FedKD (Original)            | 86.31    | 46.67      |
| FedKD + top 10% hard sample | 86.91    | 50.02      |
| FedKD + top 30% hard sample | 86.72    | 50.95      |
| FedKD + top 50% hard sample | 86.55    | 49.05      |
| FedKWAZ                     | 90.39    | 57.52      |

(1) As shown in Table 7., the performance gain from hard-sample enhancement exhibits dataset-dependent trends. On CIFAR-10, selecting the top 10% of hard samples yields the best result, while expanding the selection ratio slightly decreases performance. This indicates that the most challenging samples are highly informative, but excessive inclusion introduces redundancy and noise.

(2) On Flowers102, the best performance is achieved with the top 30% hard samples. Since this dataset has fewer training samples, an overly strict selection (e.g., 10%) limits generalization, whereas a moderate 30% ratio provides a better balance between informativeness and coverage.

(3) Overall, although selecting high-divergence samples can improve distillation, its effect is sensitive to the chosen ratio and lacks robustness. In contrast, FedKWAZ, equipped with HAPM for generating structurally diverse and perturbed samples and KDP for dynamically identifying discrepancy zones, consistently outperforms all hard-sample-based variants—achieving +4.08% gain on CIFAR-10 and +10.85% on Flowers102.

These results demonstrate that HAPM provides a more generalizable and stable mechanism for discrepancy exposure, eliminating the need for fixed selection thresholds and enabling models to better identify and assimilate knowledge weak-aware zones (KWAZ). Consequently, FedKWAZ supports more effective and fine-grained mutual distillation under heterogeneous federated learning settings.

## E.8 Correlation between HAPM Parameters and Client Properties

We investigate how the HAPM mixing parameters correlate with client properties. Eight clients (ID 0–7) are assigned private models in ascending capacity order: **4-layer CNN** < **MobileNet\_V2** < **GoogLeNet** < **ResNet18** < **ResNet34** < **ResNet50** < **ResNet101** < **ResNet152**. All clients use the same 4-layer CNN as the proxy model. On the **Flowers102** dataset (102 classes), we set the mixing strengths  $\alpha, \beta \in \{0.1, 0.5, 1.0\}$ , the spatial granularity  $g \in \{4, 16, 64\}$ , and the update frequency  $k = 30$ . Over 1000 rounds, 33 updates yield  $8 \times 33 = 264$  HAPM configurations per client, each comprising  $\{\text{SWAZ} : (\alpha^*, g^*)\}$  and  $\{\text{DWAZ} : (\beta_1^*, g_1^*), (\beta_2^*, g_2^*)\}$ .

Because KWAZ-guided mutual learning operates solely on local data, cross-client data heterogeneity has limited influence on **local parameter selection**; we therefore focus on **model architectural differences**. Based on the capacity gap between private and proxy models, clients are grouped into **Group A** (small gap, IDs 0–3; private models from 4-layer CNN to ResNet18) and **Group B** (large gap, IDs 4–7; private models from ResNet34 to ResNet152).

**Mixing strength** ( $\alpha, \beta$ ). Smaller  $\alpha$  or  $\beta$  induces stronger information mixing, amplifying inter-model discrepancies. Empirically,  $\alpha^*$ ,  $\beta_1^*$ , and  $\beta_2^*$  most frequently take **0.1** (244, 233, and 220 times, respectively), indicating that KDP often prefers smaller mixing strengths to expose cross-model gaps. Smoother settings (0.5 or 1.0) occur **26 times in Group A** and **69 times in Group B**, showing that

when private and proxy models differ substantially, even smoother mixing can still form distinct discrepancy regions that benefit transfer and alignment.

**Spatial granularity ( $g$ ).** A smaller  $g$  produces fewer (larger) patches—coarser fusion—while a larger  $g$  yields finer local mixing. The most frequent choice for  $g^*$ ,  $g_1^*$ , and  $g_2^*$  is **16** (238, 225, and 217 times, respectively), suggesting a medium granularity that balances discrepancy revelation and image recognizability. When  $g = 4$ , Group A appears **15 times** vs. Group B **38 times**; when  $g = 64$ , Group A appears **41 times** vs. Group B **18 times**. Thus, **larger structural gaps** favor **coarser mixing** (larger patch areas) to construct discrepancy regions, whereas **more similar models** benefit from **finer mixing** to explicitly stimulate subtle mismatches.

Overall, these results indicate that FedKWAZ exhibits **structure-aware behavior**: HAPM/KDP adaptively select mixing strengths and granularities according to the degree of architectural heterogeneity across clients.

### E.9 Variance and Stability Analysis of Dynamically Selected HAPM Parameters

To examine the dynamic behavior and stability of the HAPM parameters, we recorded the values of six key parameters ( $\alpha^*$ ,  $\beta_1^*$ ,  $\beta_2^*$ ,  $g^*$ ,  $g_1^*$ ,  $g_2^*$ ) selected by each client at every communication round and analyzed three representative rounds—the **30<sup>th</sup>**, **510<sup>th</sup>**, and **990<sup>th</sup>**—as summarized in Table 8.

Table 8: **Selected HAPM parameters across clients at rounds 30, 510, and 990.**

| Round | Client ID | $\alpha^*$ | $\beta_1^*$ | $\beta_2^*$ | $g^*$ | $g_1^*$ | $g_2^*$ |
|-------|-----------|------------|-------------|-------------|-------|---------|---------|
| 30    | 0         | 0.1        | 0.1         | 0.1         | 16    | 16      | 16      |
| 30    | 1         | 0.1        | 0.1         | 0.1         | 16    | 64      | 16      |
| 30    | 2         | 0.1        | 0.1         | 0.1         | 16    | 16      | 16      |
| 30    | 3         | 0.1        | 0.5         | 0.1         | 16    | 4       | 16      |
| 30    | 4         | 0.1        | 0.1         | 0.1         | 16    | 16      | 16      |
| 30    | 5         | 0.1        | 0.1         | 0.5         | 16    | 16      | 64      |
| 30    | 6         | 0.1        | 1.0         | 0.1         | 16    | 16      | 16      |
| 30    | 7         | 0.5        | 0.1         | 0.1         | 4     | 16      | 16      |
| 510   | 0         | 0.1        | 0.1         | 0.1         | 16    | 16      | 64      |
| 510   | 1         | 0.1        | 0.1         | 0.1         | 16    | 16      | 16      |
| 510   | 2         | 0.1        | 0.1         | 0.1         | 16    | 64      | 16      |
| 510   | 3         | 0.1        | 0.1         | 0.5         | 16    | 16      | 16      |
| 510   | 4         | 0.1        | 0.1         | 0.1         | 16    | 16      | 16      |
| 510   | 5         | 0.1        | 0.1         | 0.5         | 64    | 16      | 16      |
| 510   | 6         | 1.0        | 0.1         | 0.1         | 16    | 4       | 16      |
| 510   | 7         | 0.1        | 0.5         | 0.1         | 16    | 16      | 4       |
| 990   | 0         | 0.1        | 0.1         | 0.1         | 16    | 16      | 16      |
| 990   | 1         | 0.1        | 0.1         | 0.1         | 16    | 16      | 16      |
| 990   | 2         | 0.1        | 0.1         | 0.1         | 16    | 16      | 16      |
| 990   | 3         | 0.1        | 1.0         | 0.1         | 4     | 16      | 16      |
| 990   | 4         | 0.1        | 0.1         | 0.1         | 16    | 16      | 16      |
| 990   | 5         | 0.1        | 0.1         | 0.1         | 16    | 16      | 4       |
| 990   | 6         | 0.5        | 0.1         | 0.1         | 16    | 4       | 16      |
| 990   | 7         | 0.1        | 0.1         | 0.5         | 16    | 16      | 16      |

**Inter-client variance.** In the **510<sup>th</sup>** round, the variance of  $\alpha^*$  reached **0.089**, mainly because Client 6 (ResNet101, significantly more complex than the CNN proxy model) selected  $\alpha^* = 1.0$ . In such cases, clients with higher model capacities tend to prefer smoother mixing strategies to generate samples with stronger transferability. In contrast, in the **30<sup>th</sup>** round, the variance of  $g_1^*$  reached **285.75**, primarily due to Client 1 (MobileNet\_V2, a lightweight model) selecting  $g_1^* = 64$ , which is considerably higher than others. This observation suggests that when the private model is structurally close to the proxy model, the system adapts by employing finer-grained mixing strategies to better expose cross-model discrepancies.

**Intra-client variance.** Client 4 consistently selected identical values for all six parameters across rounds 30, 510, and 990, indicating strong internal stability. Similarly, most clients (Clients 0, 1, 2, and 5) maintained stable parameter selections across rounds, demonstrating that the KDP mechanism enforces a high degree of strategic consistency within each client.

Overall, FedKWAZ exhibits **low intra-client variance** (strong stability within clients) and **relatively high inter-client variance** (clear differentiation across clients). These findings confirm that FedKWAZ possesses a **structure-aware and adaptive KDP mechanism**, capable of dynamically adjusting mixing strategies according to model heterogeneity, thereby enhancing both robustness and transfer effectiveness in federated knowledge distillation.

## F Visualizations of Data Distributions

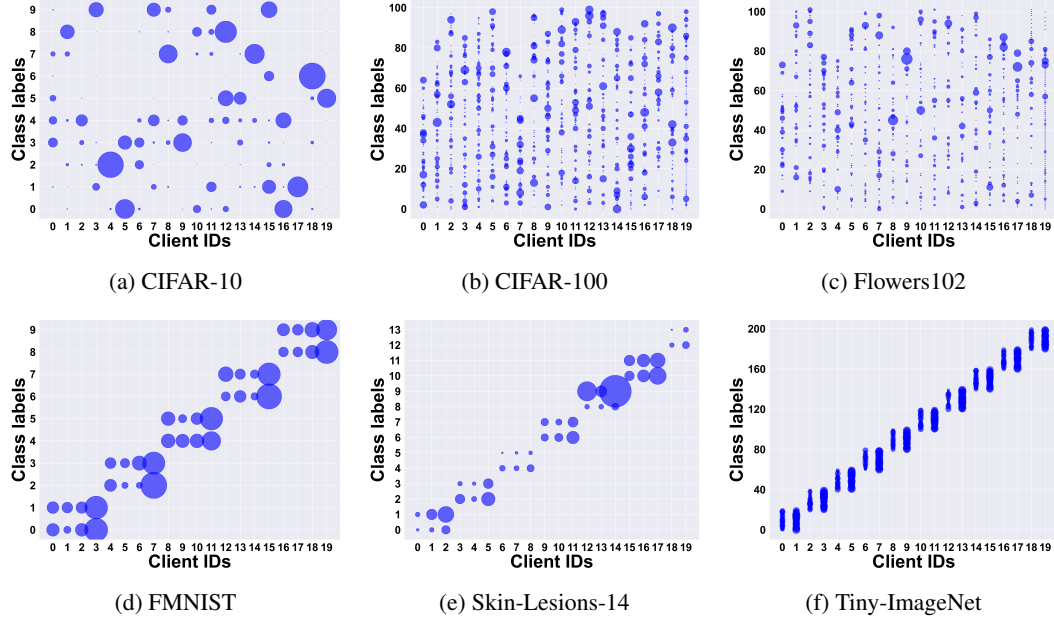


Figure 7: Top row (a–c) illustrates the **practical non-IID** scenario ( $\beta = 0.1$ ) on CIFAR-10, CIFAR-100, and Flowers102. Bottom row (d–f) shows the **pathological non-IID** setting ( $s = 2/2/20$ ) on FMNIST, Skin-Lesions-14, and Tiny-ImageNet. In each plot, each circle represents the class-wise data distribution of a client, and the size of the circle corresponds to the number of samples.

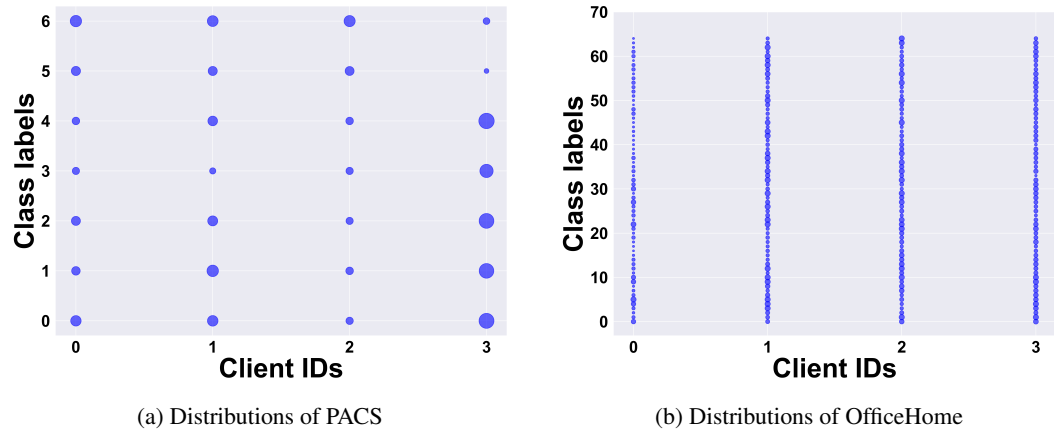


Figure 8: The data distributions of clients on PACS and OfficeHome, where each domain is assigned to an individual client. The size of each circle reflects the number of samples.

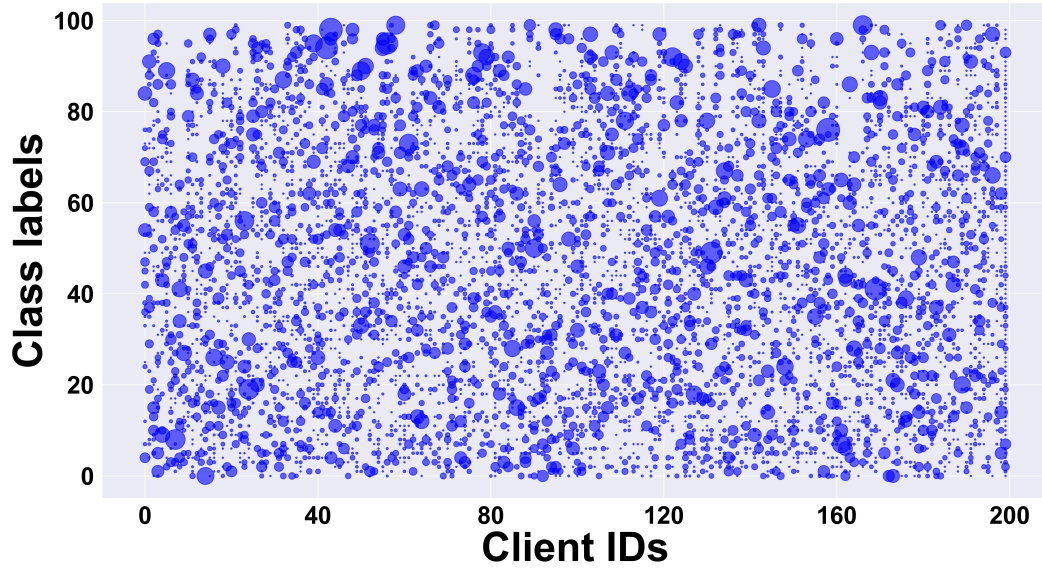
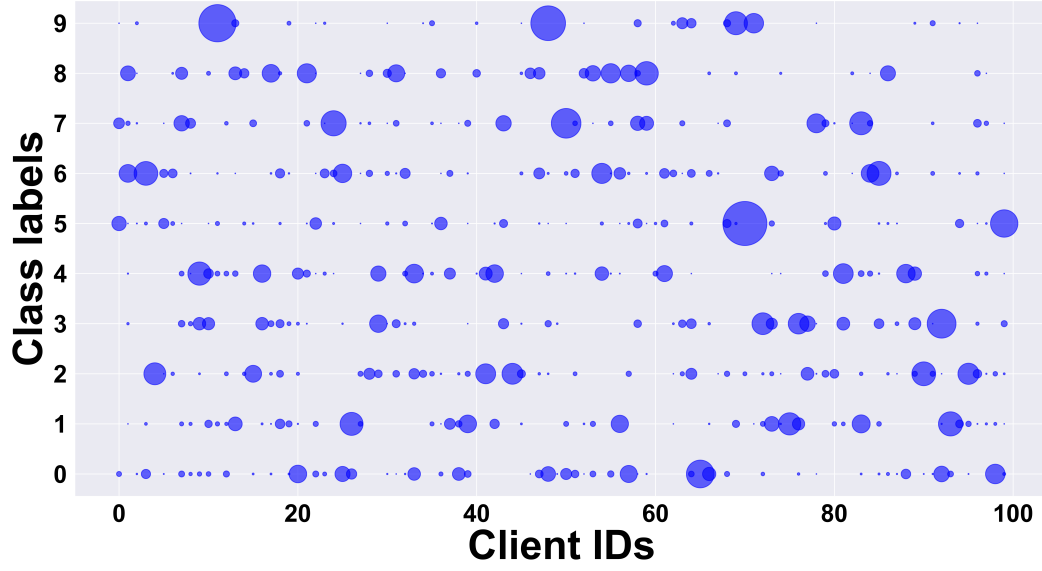


Figure 9: Data distributions of clients on Cifar10 and Cifar100 under practical settings ( $\beta = 0.1$ ). The number of clients is 100 for Cifar10 and 200 for Cifar100, respectively. The size of each circle indicates the number of samples held by each client.

## G Limitations

During the prediction distillation stage, a fixed temperature hyperparameter ( $\tau = 4.0$ ) is employed. While the empirical results demonstrate consistent performance across heterogeneous model structures and data distributions, potential remains for optimizing distillation effectiveness by dynamically adjusting the temperature in response to model complexity or input data characteristics, which may further enhance system-level efficiency.

In the default experimental configuration, the feature dimension is uniformly set to  $K = 512$ . To assess the impact of dimensionality, additional experiments are conducted with  $K = 64, 256$ , and  $1024$  on the Cifar100 dataset. As datasets differ in their demands for feature expressiveness, determining feature dimensionality in accordance with data-specific properties warrants further investigation to improve representational capacity and adaptation across tasks.

## H Broader Impacts

In the context of heterogeneous federated learning, a fine-grained mutual learning framework based on Knowledge Weak-Aware Zones (KWAZ) is proposed, yielding several broader impacts:

**Enhanced adaptability of heterogeneous models.** By explicitly modeling Semantic Weak-Aware Zones (SWAZ) and Decision Weak-Aware Zones (DWAZ), the proposed framework enables effective cross-architecture knowledge transfer. This design improves generalization under dual heterogeneity, supporting deployment in real-world scenarios with varying data distributions and model configurations.

**Reduced communication overhead and improved privacy.** By transmitting only class-level feature prototypes and prediction distributions, rather than full model parameters, the approach substantially reduces communication bandwidth consumption. Furthermore, as neither model structures nor weights are exposed, the system exhibits enhanced resistance to privacy leakage.

**Improved system efficiency via faster convergence.** The targeted refinement of KWAZ-guided distillation accelerates alignment in feature and decision spaces, enabling faster model convergence. This yields improved energy efficiency compared to conventional output-aligned distillation, contributing to the sustainability of large-scale distributed learning.

**Broadened application potential.** The architecture-agnostic design accommodates diverse client models and devices, promoting practical deployment in heterogeneous environments such as healthcare, smart manufacturing, and urban sensing systems. This broadens the real-world applicability of federated learning across domains.

## References

- [1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [3] 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] 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.
- [5] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.
- [6] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.
- [7] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
- [8] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017.
- [9] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
- [10] Sixin Zhang, Anna E Choromanska, and Yann LeCun. Deep learning with elastic averaging sgd. *Advances in Neural Information Processing Systems*, 28, 2015.