

FedARC: Adaptive Residual Compensation for Data and Model Heterogeneous Federated Learning

Author Name

Affiliation

email@example.com

1 Appendix: Supplementary Material

1.1 THEORETICAL PROOFS

Proof for Lemma 1

An arbitrary client k 's local mixed complete model $\varepsilon_t = \varepsilon_k^t \equiv \mathcal{C}_k(\varepsilon_k^t)$ can be updated by $\varepsilon_{t+1} = \varepsilon_t - \eta g_{\varepsilon,t}$ in the $(t+1)$ -th round, and following Assumption 1, we can obtain

$$\begin{aligned} \mathcal{L}_{tE+1} &\leq \mathcal{L}_{tE+0} + \langle \nabla \mathcal{L}_{tE+0}, (\varepsilon_{tE+1} - \varepsilon_{tE+0}) \rangle \\ &\quad + \frac{L_1}{2} \|\varepsilon_{tE+1} - \varepsilon_{tE+0}\|_2^2 \\ &= \mathcal{L}_{tE+0} - \eta \langle \nabla \mathcal{L}_{tE+0}, g_{\varepsilon,tE+0} \rangle \\ &\quad + \frac{L_1 \eta^2}{2} \|g_{\varepsilon,tE+0}\|_2^2. \end{aligned} \quad (1)$$

Taking the expectation of both sides of the inequality concerning the random variable ξ_{tE+0} , we obtain

$$\begin{aligned} \mathbb{E}[\mathcal{L}_{tE+1}] &\leq \mathcal{L}_{tE+0} - \eta \mathbb{E}[\langle \nabla \mathcal{L}_{tE+0}, g_{\varepsilon,tE+0} \rangle] \\ &\quad + \frac{L_1 \eta^2}{2} \mathbb{E}[\|g_{\varepsilon,tE+0}\|_2^2] \\ &\stackrel{(a)}{=} \mathcal{L}_{tE+0} - \eta \|\nabla \mathcal{L}_{tE+0}\|_2^2 \\ &\quad + \frac{L_1 \eta^2}{2} \mathbb{E}[\|g_{\varepsilon,tE+0}\|_2^2] \\ &\stackrel{(b)}{\leq} \mathcal{L}_{tE+0} - \eta \|\nabla \mathcal{L}_{tE+0}\|_2^2 \\ &\quad + \frac{L_1 \eta^2}{2} (\mathbb{E}[\|g_{\varepsilon,tE+0}\|_2^2] + \mathcal{V}(g_{\varepsilon,tE+0})) \\ &\stackrel{(c)}{=} \mathcal{L}_{tE+0} - \eta \|\nabla \mathcal{L}_{tE+0}\|_2^2 \\ &\quad + \frac{L_1 \eta^2}{2} (\|\nabla \mathcal{L}_{tE+0}\|_2^2 + \mathcal{V}(g_{\varepsilon,tE+0})) \\ &\stackrel{(d)}{\leq} \mathcal{L}_{tE+0} - \eta \|\nabla \mathcal{L}_{tE+0}\|_2^2 \\ &\quad + \frac{L_1 \eta^2}{2} (\|\nabla \mathcal{L}_{tE+0}\|_2^2 + \sigma^2) \\ &= \mathcal{L}_{tE+0} + \left(\frac{L_1 \eta^2}{2} - \eta \right) \|\nabla \mathcal{L}_{tE+0}\|_2^2 \\ &\quad + \frac{L_1 \eta^2 \sigma^2}{2}. \end{aligned} \quad (2)$$

(a), (c), (d) follow Assumption 2 and (b) follows $\mathcal{V}(x) = \mathbb{E}[x^2] - (\mathbb{E}[x])^2$.

Taking the expectation of both sides of the inequality for the model ε over E iterations, we obtain

$$\begin{aligned} \mathbb{E}[\mathcal{L}_{tE+1}] &\leq \mathcal{L}_{tE+0} + \left(\frac{L_1 \eta^2}{2} - \eta \right) \sum_{e=1}^E \|\nabla \mathcal{L}_{tE+e}\|_2^2 \\ &\quad + \frac{L_1 E \eta^2 \sigma^2}{2}. \end{aligned} \quad (3)$$

Taking the expectation of both sides of the inequality to the random variable ξ , we obtain

$$\begin{aligned} \mathbb{E}[\mathcal{L}_{(t+1)E+0}] &= \mathbb{E}[\mathcal{L}_{(t+1)E} + \mathcal{L}_{(t+1)E+0} - \mathcal{L}_{(t+1)E}] \\ &\stackrel{(a)}{\approx} \mathbb{E}[\mathcal{L}_{(t+1)E}] + \eta \|\theta_{(t+1)E+0} - \theta_{(t+1)E}\|_2^2 \\ &\leq \mathbb{E}[\mathcal{L}_{(t+1)E}] + \eta \delta^2, \end{aligned} \quad (4)$$

where (a) use the gradient of parameter variations to approximate the loss variations, i.e., $\Delta \mathcal{L} \approx \eta \cdot \|\Delta \theta\|_2^2$.

Proof for Theorem 1

Substituting Lemma 1 into the right side of Eq. (4)'s inequality, we obtain

$$\begin{aligned} \mathbb{E}[\mathcal{L}_{(t+1)E+0}] &\leq \mathcal{L}_{tE+0} + \left(\frac{L_1 \eta^2}{2} - \eta \right) \sum_{e=0}^E \|\nabla \mathcal{L}_{tE+e}\|_2^2 \\ &\quad + \frac{L_1 E \eta^2 \sigma^2}{2} + \eta \delta^2. \end{aligned} \quad (5)$$

Interchanging the left and right sides of Eq. (5), we obtain

$$\sum_{e=0}^E \|\nabla \mathcal{L}_{tE+e}\|_2^2 \leq \frac{\mathcal{L}_{tE+0} - \mathbb{E}[\mathcal{L}_{(t+1)E+0}] + \frac{L_1 E \eta^2 \sigma^2}{2} + \eta \delta^2}{\eta - \frac{L_1 \eta^2}{2}}. \quad (6)$$

Taking the expectation of both sides of the inequality over rounds $t = [0, T-1]$, we obtain

$$\sum_{e=0}^E \|\nabla \mathcal{L}_{tE+e}\|_2^2 \leq \frac{\mathcal{L}_{tE+0} - \mathbb{E}[\mathcal{L}_{(t+1)E+0}] + \frac{L_1 E \eta^2 \sigma^2}{2} + \eta \delta^2}{\eta - \frac{L_1 \eta^2}{2}}. \quad (7)$$

Let $\Delta = \mathcal{L}_{t=0} - \mathcal{L}^* > 0$, then

$\sum_{t=0}^{T-1} [\mathcal{L}_{tE+0} - \mathbb{E}[\mathcal{L}_{(t+1)E+0}]] \leq \Delta$, we can get

$$\frac{1}{T} \sum_{t=0}^{T-1} \sum_{e=0}^{E-1} \|\nabla \mathcal{L}_{tE+e}\|_2^2 \leq \frac{\frac{\Delta}{T} + \frac{L_1 E \eta^2 \sigma^2}{2} + \eta \delta^2}{\eta - \frac{L_1 \eta^2}{2}}. \quad (8)$$

Table 1: The model architectures in HtFE^{img}₈ model group. “[5 × 5, 32]” represents a convolutional layer with kernel size 5×5 and output channel 32; “2 × 2 max pool” is a max pooling layer with kernel size 2×2. “B” is short for billion.

Model	Sequentially Connected Feature Extractors	FLOPs
4-layer CNN	[5 × 5, 32], 2 × 2 max pool, [5 × 5, 64], 2 × 2 max pool, 512-d fc	0.013B
GoogLeNet	[7 × 7, 64], 3 × 3 max pool, Inception ×9, Inception ×5, Inception ×2, 1024-d fc	1.530B
MobileNetV2	[3 × 3, 32], 3 × 3 max pool, Bottleneck ×17, Bottleneck ×32, Bottleneck ×7, Bottleneck ×1, 1280-d fc	0.314B
ResNet18	[7 × 7, 64], 3 × 3 max pool, BasicBlock ×2, BasicBlock ×2, BasicBlock ×2, BasicBlock ×2, 512-d fc	0.117B
ResNet34	[7 × 7, 64], 3 × 3 max pool, BasicBlock ×3, BasicBlock ×4, BasicBlock ×6, BasicBlock ×3, 512-d fc	0.218B
ResNet50	[7 × 7, 64], 3 × 3 max pool, Bottleneck ×3, Bottleneck ×4, Bottleneck ×6, Bottleneck ×3, 2048-d fc	1.305B
ResNet101	[7 × 7, 64], 3 × 3 max pool, Bottleneck ×3, Bottleneck ×4, Bottleneck ×23, Bottleneck ×3, 2048-d fc	2.532B
ResNet152	[7 × 7, 64], 3 × 3 max pool, Bottleneck ×3, Bottleneck ×8, Bottleneck ×36, Bottleneck ×3, 2048-d fc	5.330B

If the above equation can converge to a constant ϵ , i.e.,

$$\frac{1}{T} \sum_{t=0}^{T-1} \sum_{e=0}^{E-1} \|\nabla \mathcal{L}_{tE+e}\|_2^2 \leq \frac{\frac{\Delta}{T} + \frac{L_1 \eta^2 (\sigma^2 + \delta^2)}{2}}{\eta \tilde{\mu} - \frac{L_1 \eta^2 \tilde{\mu}^2}{2}} < \epsilon, \quad (9)$$

then

$$T > \frac{\Delta}{\epsilon \left(\eta - \frac{L_1 \eta^2}{2} \right) - \frac{L_1 E \eta^2 \sigma^2}{2} - \eta \delta^2}. \quad (10)$$

Since $T > 0$, $\Delta > 0$, we can get

$$\epsilon \left(\eta - \frac{L_1 \eta^2}{2} \right) - \frac{L_1 E \eta^2 \sigma^2}{2} - \eta \delta^2 > 0. \quad (11)$$

Solving the above inequality yields

$$\eta < \frac{2(\epsilon - \delta^2)}{L_1(\epsilon + E\sigma^2)}. \quad (12)$$

Since $\epsilon, L_1, \sigma^2, \delta^2$ are all constants greater than 0, η has solutions. Therefore, when the learning rate η satisfies the above condition, any client’s local mixed complete heterogeneous model can converge. Notice that the learning rate of the local complete heterogeneous model involves $\eta_\omega, \eta_\theta, \eta_\varphi$, so it’s crucial to set reasonable them to ensure model convergence. Since all terms on the right side of Eq. (9) except for $1/T$ are constants, hence FedARC’s non-convex convergence rate is $\epsilon \sim \mathcal{O}(1/T)$.

1.2 Additional Experimental Details

Experimental Environment. To comprehensively assess the performance of FedARC, we conduct extensive experiments comparing it with 9 HtFL baselines across 4 standard benchmarks. All experiments are implemented in Pytorch and executed on a computing platform equipped with 8 NVIDIA GeForce RTX 3090 GPUs with 48GB of memory.

Algorithm 1 FedARC

Input: N , total number of clients; K , number of selected clients in one round; T , total number of rounds; η_ω , learning rate of heterogeneous local models; η_θ , learning rate of local extractors; η_φ , learning rate of representation projector.

Output: Randomly initialize the global homogeneous small model $\mathcal{G}(\theta^0)$, client local heterogeneous models $[\mathcal{F}_0(\omega_0^0), \dots, \mathcal{F}_{N-1}(\omega_{N-1}^0)]$ and local heterogeneous representation projectors $[\mathcal{P}_0(\varphi_0^0), \dots, \mathcal{P}_{N-1}(\varphi_{N-1}^0)]$.

- 1: **for** each round $t = 1, 2, \dots, T - 1$ **do**
- 2: // **Server Side:**
- 3: $\mathcal{S}^t \leftarrow$ Randomly sample K clients from N clients;
- 4: Broadcast the global homogeneous feature extractor $\theta^{ex, t-1}$ to sampled K clients;
- 5: $\theta_k^{ex, t} \leftarrow$ **ClientUpdate**($\theta^{ex, t-1}$)
- 6: /* Aggregate Local Extractor */
- 7: $\theta^{ex, t} = \sum_{k=0}^{K-1} \frac{n_k}{n} \theta_k^{ex, t}$
- 8: // **ClientUpdate:**
- 9: Receive the global homogeneous feature extractor $\theta^{ex, t-1}$ from the server;
- 10: **for** $k \in \mathcal{S}^t$ **do**
- 11: /* Local Iterative Training */
- 12: **for** $(\mathbf{x}_i, y_i) \in \mathcal{D}_k$ **do**
- 13: $\mathcal{Z}_i^{\mathcal{F}_k} = \mathcal{F}_k^{ex}(\mathbf{x}_i; \omega_k^{ex, t-1})$, $\mathcal{Z}_i^{\mathcal{G}} = \mathcal{G}^{ex}(\mathbf{x}_i; \theta_k^{ex, t-1})$;
- 14: $\mathcal{Z}_i = \mathcal{P}_k(\mathcal{Z}_i^{\mathcal{F}_k} \circ \mathcal{Z}_i^{\mathcal{G}}; \varphi_k^{t-1})$;
- 15: $\tilde{\mathcal{Z}}_i^{\mathcal{F}_k} = \mathcal{Z}_i^{1:d_1} + \tilde{\mathcal{R}}_i^{\mathcal{F}_k}$, $\tilde{\mathcal{Z}}_i^{\mathcal{G}} = \mathcal{Z}_i^{1:d_2} + \tilde{\mathcal{R}}_i^{\mathcal{G}}$;
- 16: $\hat{y}_i^{\mathcal{F}_k} = \mathcal{F}_k^{hd}(\tilde{\mathcal{Z}}_i^{\mathcal{F}_k}; \omega_k^{hd, t-1})$, $\hat{y}_i^{\mathcal{G}} = \mathcal{G}^{hd}(\tilde{\mathcal{Z}}_i^{\mathcal{G}}; \theta_k^{hd, t-1})$;
- 17: $\ell_i^{\mathcal{F}_k} = \ell_{ce}(\hat{y}_i^{\mathcal{F}_k}, y_i) + \lambda \cdot \ell_{mse}(\tilde{\mathcal{Z}}_i^{\mathcal{F}_k}, \tilde{\mathcal{Z}}_i^{g, 1:d_1})$;
- 18: $\ell_i^{\mathcal{G}} = \ell_{ce}(\hat{y}_i^{\mathcal{G}}, y_i) + \lambda \cdot \ell_{mse}(\tilde{\mathcal{Z}}_i^{\mathcal{G}}, \tilde{\mathcal{Z}}_i^{g, 1:d_2})$;
- 19: $\ell_i = \alpha_i^{\mathcal{F}_k} \cdot \ell_i^{\mathcal{F}_k} + \alpha_i^{\mathcal{G}} \cdot \ell_i^{\mathcal{G}}$.
- 20: $\omega_k^t \leftarrow \omega_k^{t-1} - \eta_\omega \nabla \ell_i$;
- 21: $\theta_k^t \leftarrow \theta_k^{t-1} - \eta_\theta \nabla \ell_i$;
- 22: $\varphi_k^t \leftarrow \varphi_k^{t-1} - \eta_\varphi \nabla \ell_i$;
- 23: **end for**
- 24: Upload updated local extractor $\theta_k^{ex, t}$ to the server.
- 25: **end for**
- 26: **end for**
- 27: **return** heterogeneous local complete models
 $[\mathcal{F}_0(\omega_0^{T-1}), \dots, \mathcal{F}_{N-1}(\omega_{N-1}^{T-1})]$.

code. Our open source code is in the anonymous github repository:

<https://anonymous.4open.science/r/FedARC-2860/main.py>

Models. As shown in Table 1, we allocate the diverse set of neural network architectures to clients. The output layer (i.e., the final fully connected layer) is configured with different dimensions based on the target dataset, such as 10 for Cifar10 and 100 for Cifar100.

Datasets. We provide 7 datasets across three modalities and three data heterogeneity scenarios. Specifically, we list all 7 datasets as follows:

1. **Cifar10:** Modality: image, Scenario: label skew, Description: 60K common images across 10 classes.
2. **Cifar100:** Modality: image, Scenario: label skew, Description: 60K common images across 100 classes.
3. **Flowers102:** Modality: image, Scenario: label skew, Description: 8K flower images across 102 classes.

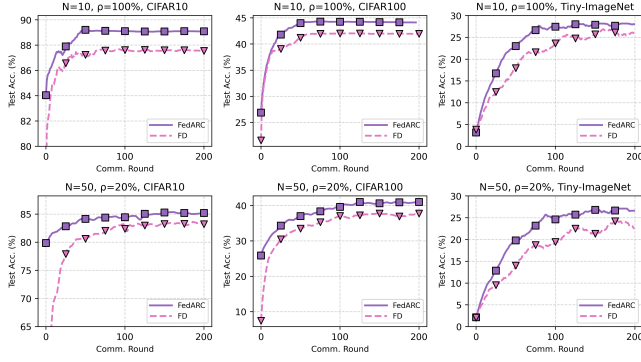


Figure 1: Average accuracy vs. communication rounds.

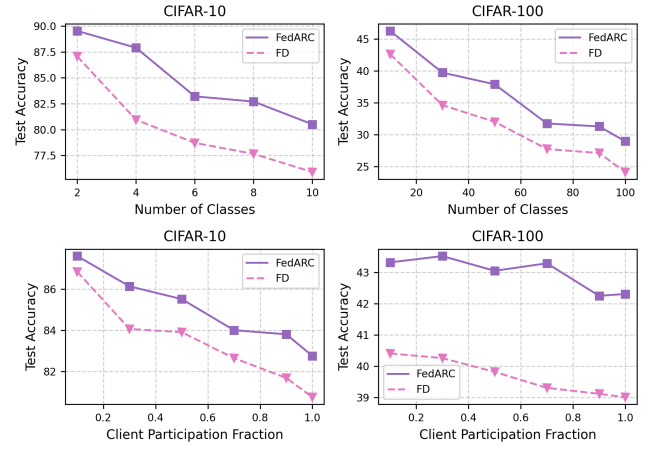


Figure 2: Robustness to non-IIDness (Class & Dirichlet).

Table 2: Test accuracy (%) on AG News in the cross-silo setting with various model heterogeneity.

	HtFE ^{txt} ₂	HtFE ^{txt} ₄	HtFE ^{txt} ₆
FD	91.63±0.12	79.06±0.21	87.89±0.15
FedProto	51.76±0.05	34.21±0.17	49.07±0.14
FedTGP	48.23±0.12	64.29±0.19	66.82±0.17
LG-FedAvg	84.43±0.08	75.46±0.23	72.19±0.18
FedGen	84.26±0.05	82.03±0.26	78.08±0.23
FedGH	86.05±0.03	77.75±0.21	79.81±0.17
pFedES	86.69±0.12	88.92±0.21	89.01±0.18
FedKD	89.23±0.02	89.16±0.26	88.71±0.04
FedMRL	85.92±0.10	88.51±0.17	89.29±0.15
FedARC	93.52±0.09	91.17±0.16	91.96±0.11

4. **Tiny-ImageNet**: Modality: image, Scenario: label skew, Description: 100K common images across 200 classes.
5. **DomainNet**: Modality: image, Scenario: feature shift, Description: 600K images across 6 domains and 345 classes.
6. **AG News**: Modality: text, Scenario: label skew, Description: 127K articles across 4 classes.

Heterogeneous Model Architectures We adopt a widely recognized strategy for selecting model architectures, prioritizing official implementations, architectural diversity, and varying representational capacities. Through an extensive survey, we have incorporated 40 heterogeneous model architectures into HtFLlib, systematically organized into 19 distinct groups. Each group is designated for a specific experiment, with X indicating the level of model heterogeneity—higher values of X correspond to greater diversity in HtFE/HtM/HtC^{dom} _{X} . The full composition of all 12 model groups is detailed below:

1. **HtFE^{img}₂**: 4-layer CNN, and ResNet18.
2. **HtFE^{img}₃**: ResNet10, ResNet18, and ResNet34.
3. **HtFE^{img}₅**: GoogLeNet, MobileNetV2, ResNet18, ResNet34, and ResNet50.
4. **HtFE^{img}₉**: ResNet4, ResNet6, ResNet8, ResNet10, ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152.
5. **Res34-HtC^{img}₄**: ResNet34 with 4 types of heads.
6. **HtFE^{img}₈-HtC^{img}₄**: HtFE^{img}₈ with 4 types of heads.
7. **HtM^{img}₁₀**: HtFE^{img}₈ plus ViT-B/16 and ViT-B/32.

1.3 Additional Experimental Results

Robustness to Non-IIDness by Class Allocation. To assess the effect of class-level heterogeneity, we systematically vary the number of classes assigned to each client under cross-device setting ($N = 50$, $\rho = 20\%$): Cifar10 uses 2/10 classes per client and Cifar100 uses 10/100. Fewer classes per client induce stronger non-IIDness. As shown in the upper panels of Figure 2, FedARC outperforms best-performing baseline (FD) across all class-imbalance levels, maintaining a clear advantage as label space per client narrows.

Robustness to Participation Rates. We test the robustness of FedARC and FD against different client participant rates $\rho \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$ with $N = 100$ on Cifar10/100 (non-IID: 2/10 and 10/100). Across all ρ , FedARC remains superior to FD, with larger margins on the more challenging Cifar100 (Figure 2). Notably, as ρ increases, more clients per round provide more IID local data, steering updates toward a globally averaged model—improving generalization while diluting personalization.

Impact of Model Heterogeneity. From Table 2, we note three trends. (1) For methods that share only part of the parameters, accuracy typically declines as architectural heterogeneity increases. (2) By contrast, mutual-distillation and prototype-sharing approaches do not show a strictly monotonic drop with greater heterogeneity. (3) Beyond heterogeneity itself, the effectiveness of feature extraction strongly affects prototype-based methods: on text, FedProto and FedTGP lag behind their image-task counterparts. This indicates that, in NLP, architectural differences lead to divergent processing and context-modeling behaviors, making a unified feature space harder to align across clients. Consequently, aligning in the logit space is generally more reliable and efficient than direct feature-space alignment for text.

Table 3: Test accuracy (%) on Cifar100 in the cross-silo setting using HtFE^{img}_8 with different values of β . The results in “()” indicate the total number of converged rounds. We omit error bars here due to limited space.

	$\beta = 0.01$	$\beta = 0.1$	$\beta = 0.5$	$\beta = 1$
FD	68.10(322)	42.67(208)	23.31(153)	17.25(269)
FedProto	60.26(562)	36.12(556)	19.41(581)	12.81(391)
FedTGP	67.82(226)	40.69(211)	22.87(217)	18.30(263)
LG-FedAvg	66.46(172)	40.16(184)	21.93(265)	15.81(138)
FedGen	66.16(161)	38.68(158)	21.53(141)	15.45(158)
FedGH	66.32(158)	41.61(219)	21.82(228)	15.57(196)
pFedES	68.53(288)	42.37(201)	23.28(175)	17.83(322)
FedKD	65.93(279)	41.91(191)	22.35(160)	18.21(279)
FedMRL	68.92(188)	42.57(171)	23.41(145)	18.11(421)
FedARC	72.52(140)	44.21(147)	25.61(132)	20.17(164)

Table 4: Test accuracy (%) on Cifar100 using HtFE^{img}_8 with 50/100/200 clients under partial participation (ρ).

	$\rho = 50\%$			$\rho = 10\%$
	50 Clients	100 Clients	200 Clients	100 Clients
FD	38.48±0.31	35.65±0.23	32.07±0.09	40.41±0.51
FedProto	20.20±0.48	18.81±0.51	15.25±0.42	20.76±0.92
FedTGP	36.71±0.21	<u>35.87±0.28</u>	27.81±0.63	29.53±0.51
LG-FedAvg	36.42±0.11	35.08±0.42	27.78±0.08	<u>37.87±0.26</u>
FedGen	35.02±0.23	33.54±0.23	26.91±0.21	33.34±0.48
FedGH	<u>36.89±0.08</u>	<u>35.21±0.14</u>	<u>30.14±0.35</u>	37.43±0.76
pFedES	<u>37.92±0.29</u>	35.62±0.62	<u>34.32±0.52</u>	39.49±0.32
FedKD	37.62±0.21	35.08±0.56	33.23±0.48	35.68±0.25
FedMRL	37.89±0.21	<u>36.77±0.55</u>	31.99±0.72	<u>39.52±0.29</u>
FedARC	40.56±0.10	38.12±0.35	36.56±0.21	41.74±0.18

Impact of Data Heterogeneity. Table 3 reports accuracy and convergence rounds (in parentheses) on Cifar100 with HtFE^{img}_8 across different β . Prototype-sharing methods degrade markedly and often require many more rounds under stronger heterogeneity (e.g., FedProto), consistent with the fact that aggregating class means assumes a roughly aligned feature space—under label/feature skew the global prototypes become biased and less representative of local decision boundaries. By contrast, methods with server-side regularization or synthetic signals (e.g., FedTGP’s margin-aware anchors; FedGen’s generator-based alignment) tend to maintain more stable convergence as they partially decouple local updates from idiosyncratic client data. More broadly, theory shows that non-IID partitions induce gradient dissimilarity and client drift, slowing or destabilizing FedAvg-style training and increasing the communication rounds needed to reach a target accuracy; normalization/variance-reduction remedies were proposed exactly to counter this effect. Within this landscape, FedARC retains clear accuracy margins across all β while converging among the fastest: its projection-based fusion plus alignment explicitly reduces cross-client representation mismatch, which mitigates drift and lowers the rounds needed to reach high accuracy.

Impact of Client Participation. We benchmark all methods with 50, 100, and 200 clients under partial participation ($\rho < 100\%$). From Tab. 4: (1) Accuracy drops for every baseline as the client pool grows, because partitioning Cifar100

across more clients leaves fewer samples per client, weakening local updates and, in turn, aggregation. (2) FedMRL benefits from jointly training a small global model with local models and therefore degrades less at low ρ , alleviating under-aggregation. (3) FedTGP remains competitive at 100 clients with $\rho = 50\%$, but deteriorates markedly at $\rho = 10\%$, where per-round participation is too limited to maintain reliable prototype guidance. Crucially, FedARC not only preserves its edge but widens it as the scenario becomes more dynamic. Despite larger participation size and unstable participation, FedARC remains the top performer. This indicates that our first-order semantic anchors—re-estimated every round, smoothed with a light EMA, and coupled with residual compensation and λ warm-up—do not lag or drift in highly dynamic environments.

LLM Usage Disclosure

We used large language models solely to aid with language editing (grammar, clarity, and style) on author-written drafts. No model was used to generate technical content, claims, code, analyses, figures, or references; all scientific contributions are by the authors, who verified all text.