This is the peer reviewd version of the followng article:

Intrinsic Training Signals for Federated Learning Aggregation / Fiorini, C., Mosconi, M., Buzzega, P., Salami, R., Calderara, S.. - (2025). (23rd International Conference on Image Analysis and Processing Roma 19/09/2025).

Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

11/10/2026 14:00

(Article begins on next page)

arXiv:2507.06813v2 [cs.LG] 15 Sep 2025

Intrinsic Training Signals for Federated Learning Aggregation

Cosimo Fiorini, Matteo Mosconi, Pietro Buzzega, Riccardo Salami, and Simone Calderara

AImageLab - University of Modena and Reggio Emilia, Italy
name.surname@unimore.it

Abstract. Federated Learning (FL) enables collaborative model training across distributed clients while preserving data privacy. While existing approaches for aggregating client-specific classification heads and adapted backbone parameters require architectural modifications or loss function changes, our method uniquely leverages intrinsic training signals already available during standard optimization. We present LIVAR (Layer Importance and VARiance-based merging), which introduces: *i)* a variance-weighted classifier aggregation scheme using naturally emergent feature statistics, and *ii)* an explainability-driven LoRA merging technique based on SHAP analysis of existing update parameter patterns. Without any architectural overhead, LIVAR achieves state-of-the-art performance on multiple benchmarks while maintaining seamless integration with existing FL methods. This work demonstrates that effective model merging can be achieved solely through existing training signals, establishing a new paradigm for efficient federated model aggregation. The code is available at https://github.com/aimagelab/fed-mammoth.

Keywords: Federated Learning · Model Merging · Explainability

1 Introduction

The limited ability of deep models to effectively learn from sparse and distributed real-world data has necessitated the development of Federated Learning (FL), a paradigm in which clients train on their local data distributions and share model updates with a central server. The server aggregates these updates without accessing raw local data, thereby preserving privacy. FL has found practical applications in diverse domains, such as manufacturing, where factories collaborate to build a unified model without sharing proprietary sensor data [20], and healthcare, where hospitals cannot exchange sensitive patient information [24]. In recent years, Parameter-Efficient Fine-Tuning (PEFT) [3] has emerged as a training paradigm well-suited to FL. PEFT techniques leverage pre-trained architectures [21] to reduce the number of trainable parameters while maintaining performance comparable to full fine-tuning. Beyond computational efficiency, PEFT is particularly advantageous in FL, as it minimizes communication overhead by transmitting fewer parameters.

Recent studies employing pre-trained architectures highlight distinct contributions of fine-tuning the backbone versus the final classification layer, with evidence suggesting adapting the classifier plays a more critical role in model performance [32]. Consequently, some FL methods employ specialized strategies for tuning the classifier [17] while aggregating backbone parameters via simple averaging [19]. Others utilize PEFT techniques, like LoRA [7], to fine-tune the backbone [25] and subsequently merge updated vectors. Research explored various approaches for determining module aggregation coefficients [29,30], including learnable weighting mechanisms [28] and closed-form solutions [22].

Alternatively, other approaches employ separate training and aggregation solutions for the backbone and classification head [23]. While we consider this decoupled strategy superior, we also advocate for exploring more lightweight solutions for both classifier head merging and backbone module aggregation. We argue that leveraging naturally available training signals can establish both efficient and effective Federated Learning baselines.

Building upon these foundations, we present LIVAR, a novel Federated Learning framework that implements a decoupled approach for aggregating both backbone parameters and classification heads. For classifier merging, we propose variance-weighted heads, where each client's class-specific head is weighted by the variance of features from correctly predicted training samples of that class. Intuitively, higher variance indicates richer class-specific information, and this measure is obtained effortlessly during local training.

For backbone adaptation, LIVAR employs LoRA modules, integrating them with a novel explainability-driven aggregation scheme. We're the first to propose applying SHAP values [15] to determine optimal merging coefficients for federated LoRA modules. Aggregation weights are derived from layer-wise gradients and parameter update patterns during client training; larger cumulative updates signify more substantial adaptations of specific network components. These naturally emergent signals – available without architectural modifications – serve as proxies for each client's contribution to particular model parameters. To implement this, we leverage a proxy dataset to construct an explainability curve that directly maps these adaptation patterns to their optimal merging coefficients.

We extensively evaluate LIVAR on in-distribution and out-of-distribution datasets w.r.t. to pre-training. Our method achieves state-of-the-art performance in all in-distribution settings and remains competitive otherwise. Furthermore, we demonstrate that our classifier weighting scheme remains compatible with existing head-specific strategies, enabling seamless integration with these approaches. Beyond FL, our work highlights the broader potential of explainability-driven module merging, opening new research directions in model aggregation.

In summary, our contributions are as follows:

- We propose a variance-weighted aggregation strategy for classification heads.
- We conceive explainability-driven merging coefficients for the federated aggregation of LoRA modules.
- We propose LIVAR, achieving state-of-the-art results across multiple benchmarks while maintaining seamless integration with existing approaches.

2 Related Works

Federated Learning addresses the challenges of training models in a distributed environment, where local data are private and cannot be exchanged. Federated Averaging (FedAvg) [19] established the foundation by averaging client model weights, yet its simplicity makes it vulnerable to highly heterogeneous data streams, which inherently introduce local client divergence. To overcome this challenge, techniques such as Scaffold [10] introduce control variates that compensate for local updates, thereby reducing client drift. Other approaches include FedProx [14], which incorporates a proximal term to constrain the deviation between local and global models, and MOON [13], which employs contrastive loss to align the representations learned by local and global models. Similarly, CCVR [17] enhances the classifier on the server side by rebalancing it with synthetic features, thereby stabilizing updates across heterogeneous clients and promoting convergence. FedProto [26] leverages prototypical representations to regularize the local training process, enabling the maintenance of class-specific information, while GradMA [16] uses a gradient-memory mechanism to guide local update processes by incorporating historical gradient information, thereby stabilizing update and accelerating model convergence.

Model Merging methodologies have emerged as powerful alternatives to conventional weight averaging in FL systems. Fisher Averaging [18] leverages the Fisher Information Matrix to weight client contributions, thus prioritizing more informative updates during aggregation. RegMean [9] presents a closed-form solution for merging model weights, enabling the identification of a global model that best approximates the responses of the clients, demonstrating strong potential for federated settings. Following the introduction of LoRA (Low-Rank Adaptation) [7], a growing body of research has emerged to address the challenge of optimal coefficient selection for module merging [8, 28–30], including several approaches specifically designed for Federated Learning scenarios [22]. While these methods achieve promising results, they commonly rely on auxiliary architectural elements, computationally expensive procedures, or customized loss functions. In contrast, our methodology introduces a more streamlined solution that utilizes naturally emerging training signals to effectively aggregate both the LoRA-adapted backbone and the classification head without requiring additional components.

3 Preliminaries

LoRA. Low-Rank Adaptation [7] is an efficient parameter adaptation technique that significantly reduces the number of trainable parameters in deep neural networks. For a given layer with weight matrix $W \in \mathbb{R}^{d\times k}$, LoRA decomposes the weight update ΔW into two trainable low-rank matrices: $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$, where the rank $r \ll min(d, k)$. The adapted weights are computed as:

$$W' = W + BA \tag{1}$$

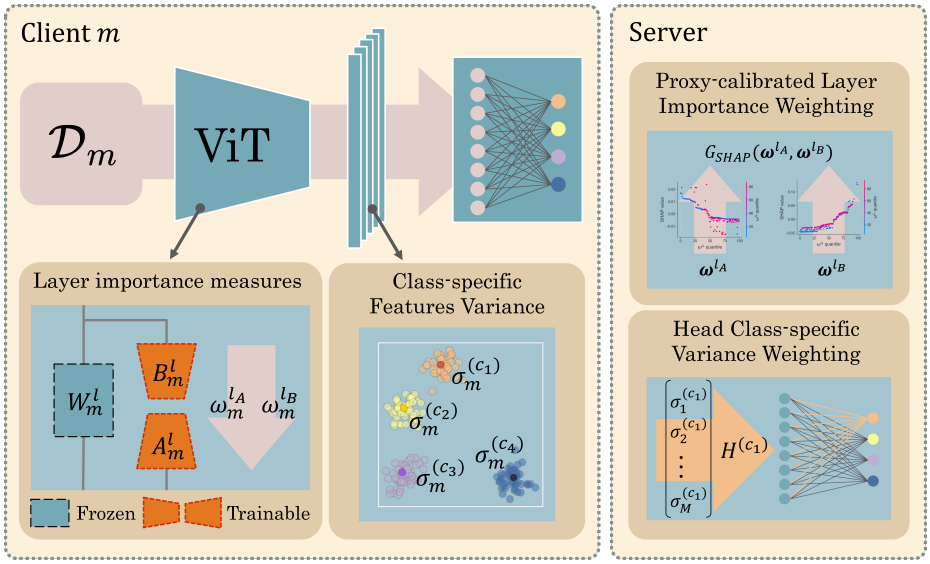

Fig. 1: LIVAR employs a dual weighting strategy architecture. The left diagram depicts client-side information collection during training, the right diagram shows the distinct weighting strategies used on the server side.

During adaptation, the original weights W remain frozen while only the low-rank matrices A and B are updated. This decomposition reduces the number of trainable parameters from $d \times k$ to $r \times (d + k)$, resulting in significant memory savings compared to full fine-tuning.

Problem Setting. We address a C-class classification problem in Federated Learning where data are distributed non-IID across M clients. Let $\mathcal{D}_m$ represent the data partition for client m. The training process consists of sequential communication rounds; at each round, clients perform local optimization for a specified number of epochs, and then the server aggregates the optimized parameters before redistributing them to clients. Let f_θ denote the global model parameterized by θ. The training procedure aims to find the optimal parameter set θ that minimizes the loss across the entire dataset:

$$\underset{\theta}{\text{minimize}} \quad \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}_{(x,y)\sim\mathcal{D}_m} \mathcal{L}(f_\theta(x), y) \tag{2}$$

4 LIVAR

We present LIVAR, a Federated Learning framework with distinct aggregation strategies for the backbone and the classification head. Each client m maintains parameters $\theta_m = \{W_m, H_m\}$, where W_m represents the backbone weights and H_m denotes the classification head.

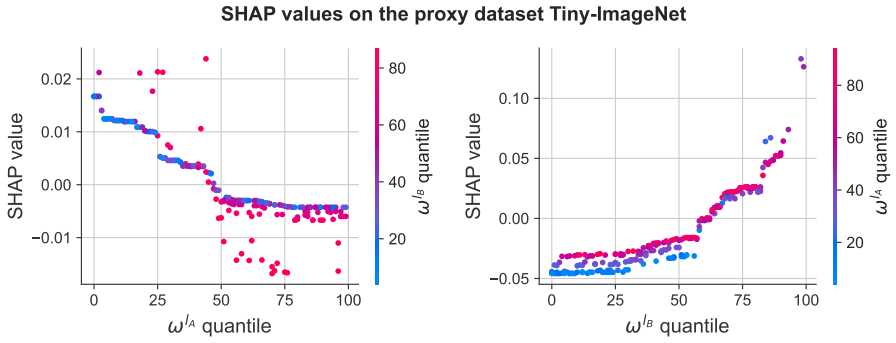

Fig. 2: SHAP values represent the impact of $\omega_m^{l_A}$ and $\omega_m^{l_B}$ features in the prediction of optimal merging coefficients λ_m^l. Values on x axis are expressed in quantiles with respect to the values of all clients and all layers.

For the backbone, we employ Low-Rank Adaptation with our novel *Proxy-Calibrated Layer Importance Weighting* mechanism, using explainability-driven coefficients to merge client updates. The classification head uses *data-variance-aware* merging that automatically adapts to client data heterogeneity, ensuring robustness under non-IID distributions. The dual merging strategy is depicted in Figure 1.

4.1 Proxy-Calibrated Layer Importance Weighting

Each client m learns L low-rank adaptations $\Delta W_m = \{\Delta W_m^1, \ldots, \Delta W_m^L\}$, where each adaptation $\Delta W_m^l = B_m^l A_m^l$ follows the standard LoRA decomposition. The server aggregates these client updates through a layer-wise weighted combination:

$$\Delta W^l = \sum_{m=1}^{M} \alpha_m^l \cdot \Delta W_m^l \tag{3}$$

where α_m^l represents the layer-specific merging coefficients that constitute the target of our estimation procedure.

These α_m^l coefficients are derived through a three-stage process. First, we establish a ground truth reference using a proxy dataset (*i.e.*, Tiny-ImageNet [12]) by training five federated clients in parallel with a centralized model trained on the full dataset. This process yields the reference parameters $\Delta W_{\text{joint}}^l$ of the centralized model. The fundamental premise is that optimal aggregation should minimize the distance between federated combinations and centralized training outcomes. Formally, for each layer l, we solve the constrained regression problem:

$$\underset{\lambda_m^l}{\text{minimize}} \quad \|\Delta W_{joint}^l - \sum_{m=1}^{5} \lambda_m^l \Delta W_m^l\|_2^2 \quad \text{subject to} \quad \lambda_m^l \geq 0 \tag{4}$$

Algorithm 1 LIVAR training for a single round

Input: M clients, L LoRA layers, $G_{SHAP}(\cdot)$ proxy-calibrated function, C classes.
Client side:
for client $m \in \{1, \ldots, M\}$ **do**
  $\Delta W_m \leftarrow (training)$ ▷ Collect $(\omega_m^{l_A}, \omega_m^{l_B})$ during training
  Collect $\sigma_m^{(c)}$ over well classified examples
  Send $(\omega_m^{l_A}, \omega_m^{l_B})$ and $\sigma_m^{(c)}$ to the server
end for
Server side:
for layer $l \in \{1, \ldots, L\}$ **do**
  $\alpha_m^l \leftarrow$ use Equation (5)
  $\Delta W^l \leftarrow$ use Equation (3) ▷ Proxy-Calibrated Layer Importance Weighting
end for
for class $c \in \{1, \ldots, C\}$ **do**
  $\Sigma^{(c)} = \sum_m (\sigma_m^{(c)})$
  $H^{(c)} \leftarrow$ use Equation (6) ▷ Head Class-Specific Variance Weighting
end for

Second, we generalize this solution by establishing a mapping from trainable parameters to optimal coefficients. Our method builds on the key insight that a layer's contribution to loss variation effectively captures its importance during server-side aggregation, providing a reliable basis for estimating the optimal merging coefficients λ_m^l. Following [31], we compute layer importance measures $\omega_m^{l_A}$ and $\omega_m^{l_B}$ as the running sums of Hadamard products between gradients and parameter updates for matrices A_m^l and B_m^l respectively. To find the relationship between layer importance scores $\{\omega_m^{l_A}, \omega_m^{l_B}\}$ and actual coefficients λ_m^l, we train a Random Forest regressor [1] to predict λ_m^l from the importance signals. This enables a robust, explainable, and efficient estimation of coefficients on new datasets, without relying on a centralized (joint) reference model.

Third, we employ SHAP analysis [15] to interpret the trained model (see Figure 2), revealing distinct roles of the A and B matrices – the former initialized via Kaiming uniform distribution [4] to explore directions, the latter initialized to zeros to control direction utilization. The SHAP values are discretized through a quantile-based function G_{SHAP} to yield the final coefficients:

$$\alpha_m^l = G_{SHAP}(\omega_m^{l_A}, \omega_m^{l_B}) \tag{5}$$

Additional implementation details regarding the quantile-based discretization procedure are provided in Section A.

Beyond the proxy dataset. The function $G_{SHAP}(\cdot)$, which estimates merging coefficients from importance signals $\omega_m^{l_A}$ and $\omega_m^{l_B}$ observed during training, enables federated aggregation across arbitrary datasets. Our quantile-based discretization ensures robust generalization beyond the proxy dataset used for calibration. We validate this generalization capability through a posteriori analysis of SHAP value distributions across multiple benchmarks in Section 5.4.

Table 1: Comparison of approaches on in-distribution datasets.

	CIFAR-100			ImageNet-R		
Distribution β	1.0	0.5	0.2	1.0	0.5	0.2
FedAvg	93.2	92.93	91.56	84.38	81.68	78.73
RegMean	93.36	93.09	92.11	84.12	83.35	81.18
FisherAvg	93.4	93.34	92.22	84	83.78	82.67
FedProto	92.99	92.9	89.88	84.27	82.2	76.97
CCVR	93.27	93.38	92.35	84.87	84.17	82.8
LIVAR	**93.57**	**93.58**	**92.92**	**85.45**	**84.35**	**83.05**

For server-side aggregation, the coefficients α_m^l computed via Equation (5) are applied to combine client updates ΔW_m^l, as in Equation (3), with each client's importance measures $\omega_m^{l_A}$ and $\omega_m^{l_B}$ derived from their local training process.

4.2 Head Class-specific Variance Weighting

The classification head aggregation component of LIVAR operates on the principle that feature-space variance correlates with class representation quality. At the end of client-side training, we compute the mean variance $\sigma_m^{(c)}$ of pre-logits features for correctly classified examples of each class c, serving as a proxy for how comprehensively each client represents the class.

At the server, aggregation weights are derived from normalized variance contributions across clients. For each class c, the server first computes the total variance $\Sigma^{(c)} = \sum_m \sigma_m^{(c)}$ across all clients, then performs the weighted aggregation as:

$$H^{(c)} = \sum_m \frac{\sigma_m^{(c)}}{\Sigma^{(c)}} H_m^{(c)} \tag{6}$$

This variance-weighted scheme prioritizes classification heads whose training data exhibits broader coverage of each class's feature distribution. The complete training procedure is formally specified in Algorithm 1.

5 Experimental Studies

5.1 Evaluation Settings

We conducted a comprehensive evaluation of LIVAR on both in-distribution datasets (CIFAR-100 [11], ImageNet-R [6]) and out-of-distribution benchmarks (EuroSAT [5], CUB-200 [27]) relative to the pre-training domain. Data was distributed across 10 clients using a *distribution-based* label imbalance setting governed by a Dirichlet distribution with parameter β. We evaluated all methods under three distribution-based imbalance scenarios: $\beta \in \{0.2, 0.5, 1.0\}$, where

Table 2: Comparison of approaches on out-of-distribution datasets.

	EuroSAT			CUB-200		
Distribution β	1.0	0.5	0.2	1.0	0.5	0.2
FedAvg	98.35	**98.36**	97.43	73.51	78.01	61.98
RegMean	98.44	98.28	97.47	74.85	70.94	59.8
FisherAvg	**98.49**	97.98	96.86	73.11	75.16	75.92
FedProto	98.31	98.15	97.11	79.05	74.51	62.79
CCVR	98.44	**98.36**	**97.83**	85.42	84.79	**84.66**
LIVAR	98.47	98.17	97.53	**86.24**	**85.73**	82.86

lower values indicate greater data heterogeneity. For all experiments, we evaluated model performance using a global test set.

We benchmark LIVAR against five state-of-the-art approaches spanning complementary research directions: FedAvg [19], CCVR [17], and FedProto [26] represent federated learning methods, while RegMean [9] and Fisher Merging [18] constitute established model merging techniques. Our implementation leverages a ViT-B/16 [2] backbone pre-trained on ImageNet-21K [21] across all methods. For LIVAR, we applied LoRA adaptation to all *qkv*, *mlp*, and *multi-head attention* projection layers. For all datasets, we standardized the training protocol with 5 communication rounds, each comprising 5 local epochs.

5.2 Results of LIVAR

We report comparative results in terms of average accuracy across all evaluated methods.

In distribution. As reported in Table 1, for both in-distribution datasets, we observe consistent trends: LIVAR achieves state-of-the-art results across all β settings, while CCVR consistently ranks second. For both datasets, higher degrees of distribution imbalance yield more pronounced differences in performance – while LIVAR and CCVR maintain their leading performance, the model merging approaches follow closely, and other methods fail to remain robust under these harder conditions.

Out of distribution. Table 2 shows that the EuroSAT dataset proves less informative for evaluating Federated Learning with ViT-based methods, as all models achieve similar accuracy levels – likely due to the dataset's limited 10-class structure. In contrast, CUB-200 presents a more challenging benchmark where LIVAR and CCVR significantly outperform competing approaches. LIVAR achieves state-of-the-art results for moderate data imbalance ($\beta \in \{1, 0.5\}$), with CCVR consistently ranking second. However, under extreme imbalance ($\beta = 0.2$), CCVR surpasses LIVAR, demonstrating the advantages of its classifier rebalancing strategy over our variance-based approach.

Method composability. We investigate LIVAR's compatibility with existing federated learning techniques through integration with CCVR on CUB-200 – a

Table 3: Evaluation of the combined performance of LIVAR and CCVR [17] on CUB-200 across varying β-distribution parameter.

Distribution β	1.0	0.5	0.2
CCVR	85.42	84.79	84.66
LIVAR	86.24	85.73	82.86
LIVAR +CCVR	**86.52**	**86.42**	**86.37**

Table 4: Ablation study on LIVAR components.

α^l	$\sigma^{(c)}$	Avg (std)
✗	✗	79.16 (0.16)
✓	✗	79.81 (0.12)
✗	✓	90.63 (0.07)
✓	✓	90.90 (0.07)

dataset representing a significant domain shift from ViT's pre-training distribution. While CCVR alone outperforms LIVAR in the extreme imbalance setting ($\beta = 0.2$), we demonstrate their complementary strengths when combined: applying LIVAR's layer-specific aggregation followed by CCVR's server-side calibration with synthetic data yields superior performance to either method in isolation. As shown in Table 3, this synergy highlights how LIVAR's feature variance head weighting provides foundational benefits that enhance subsequent refinement techniques like CCVR. This composability represents a significant advantage, as it allows practitioners to combine multiple techniques to address the unique challenges of different federated learning scenarios.

5.3 Ablation Study on LIVAR Components

We systematically evaluate LIVAR's components by measuring Average Accuracy and standard deviation on CIFAR-100 with extreme label imbalance ($\beta = 0.05$). Starting with a baseline of FedAvg-aggregated ViT-B/16 parameters – including both LoRA residuals ΔW^l and the classification head H – Table 4 demonstrates progressive improvements through incremental enhancements. The Proxy-Calibrated Layer Importance Weighting (α^l) achieves statistically significant accuracy gains beyond the standard deviation across five runs (reported in parentheses in Table 4). Subsequent incorporation of Head Class-Specific Variance Weighting ($\sigma^{(c)}$) delivers substantially greater improvement, corroborating the classification head's critical role. The full LIVAR framework combines these components synergistically to achieve peak performance.

5.4 A Posteriori Analysis of SHAP Results

To verify the generalizability of our proxy-calibrated SHAP analysis explained in Section 4.1, we repeated the calibration procedure across all four datasets: CIFAR-100, ImageNet-R, EuroSAT, and CUB-200. Importantly, this analysis does not involve recalibrating the $G_{SHAP}(\cdot)$ function for each specific dataset – since such per-dataset calibration would be impossible in a federated setting where the joint data distribution is inaccessible – but instead demonstrates that the original proxy-based calibration effectively generalizes to different data domains.

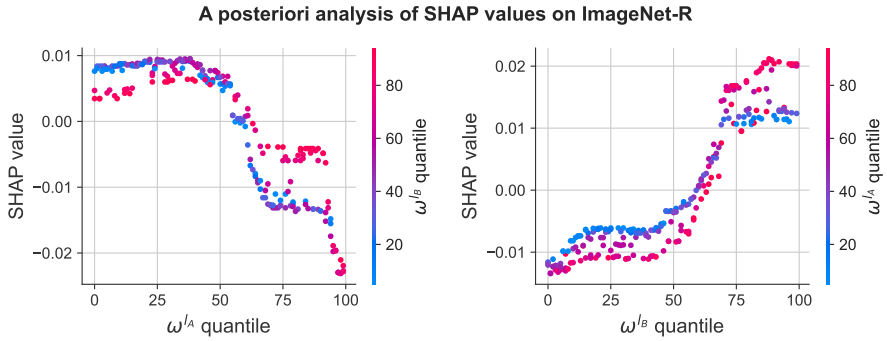

Fig. 3: A posteriori analysis on $\omega_m^{l_A}$ and $\omega_m^{l_B}$ impact in predicting λ_m^l coefficients on ImageNet-R. The trend of both features' impact is consistent with the one observed in the proxy dataset.

Following the same methodology that produced the results in Figure 2, we generated comparable analyses for each dataset. As shown in Figure 3 for ImageNet-R, the SHAP value trends remain consistent: ω^{l_A} exhibits decreasing or stable influence with increasing feature values, while ω^{l_B} maintains its clear increasing trend across all datasets. These consistent patterns (fully documented in Section B) confirm the proxy dataset's representativeness, despite variations in specific impact magnitudes.

The observed consistency, coupled with the variations in absolute values, provides empirical justification for our ω^l discretization approach. This design choice ensures LIVAR's robustness when applied to diverse datasets while effectively preventing overfitting to the specific characteristics of the proxy dataset used for calibration.

6 Conclusions

We introduce LIVAR, a Federated Learning framework that advances model aggregation through two novel strategies. For backbone adaptation, we develop an explainability-driven merging procedure for LoRA modules based on training-emerging signals, while our classifier aggregation employs feature variance to weight client contributions. The method demonstrates state-of-the-art performance across multiple benchmarks while maintaining seamless compatibility with existing approaches.

Looking ahead, we identify promising opportunities to extend this work through local explainability techniques for coefficient estimation, potentially removing the need for proxy datasets and enabling more dataset-specific merging. We believe these directions will not only benefit Federated Learning but could also influence broader model fusion research, particularly for explainable model adaptation.

References

1. Breiman, L.: Random forests. Machine learning (2001)
2. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Int. Conf. Learn. Represent. (2021)
3. Fu, Z., Yang, H., So, A.M.C., Lam, W., Bing, L., Collier, N.: On the effectiveness of parameter-efficient fine-tuning. In: AAAI (2023)
4. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Int. Conf. Comput. Vis. (2015)
5. Helber, P., Bischke, B., Dengel, A., Borth, D.: Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (2019)
6. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: Int. Conf. Comput. Vis. (2021)
7. Hu, E.J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. In: Int. Conf. Learn. Represent. (2022)
8. Huang, C., Liu, Q., Lin, B.Y., Pang, T., Du, C., Lin, M.: Lorahub: Efficient cross-task generalization via dynamic lora composition. arXiv preprint arXiv:2307.13269 (2023)
9. Jin, X., Ren, X., Preotiuc-Pietro, D., Cheng, P.: Dataless knowledge fusion by merging weights of language models. In: Int. Conf. Learn. Represent. (2023)
10. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: Scaffold: Stochastic controlled averaging for federated learning. In: Int. Conf. Mach. Learn. (2020)
11. Krizhevsky, A.: Learning multiple layers of features from tiny images. Master's thesis, University of Tront (2009)
12. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N (2015)
13. Li, Q., He, B., Song, D.: Model-contrastive federated learning. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021)
14. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems (2020)
15. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. Adv. Neural Inform. Process. Syst. (2017)
16. Luo, K., Li, X., Lan, Y., Gao, M.: Gradma: A gradient-memory-based accelerated federated learning with alleviated catastrophic forgetting. In: IEEE Conf. Comput. Vis. Pattern Recog. (2023)
17. Luo, M., Chen, F., Hu, D., Zhang, Y., Liang, J., Feng, J.: No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. Adv. Neural Inform. Process. Syst. (2021)
18. Matena, M.S., Raffel, C.A.: Merging models with fisher-weighted averaging. Adv. Neural Inform. Process. Syst. (2022)
19. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: International Conference on Artificial Intelligence and Statistics (2017)

20. Pruckovskaja, V., Weissenfeld, A., Heistracher, C., Graser, A., Kafka, J., Leputsch, P., Schall, D., Kemnitz, J.: Federated learning for predictive maintenance and quality inspection in industrial applications. In: 2023 Prognostics and Health Management Conference (PHM). IEEE (2023)
21. Ridnik, T., Ben-Baruch, E., Noy, A., Zelnik-Manor, L.: Imagenet-21k pretraining for the masses. In: Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2021)
22. Salami, R., Buzzega, P., Mosconi, M., Bonato, J., Sabetta, L., Calderara, S.: Closed-form merging of parameter-efficient modules for federated continual learning. In: Int. Conf. Learn. Represent. (2025)
23. Salami, R., Buzzega, P., Mosconi, M., Verasani, M., Calderara, S.: Federated class-incremental learning with hierarchical generative prototypes. arXiv preprint arXiv:2406.02447 (2024)
24. Sheller, M.J., Edwards, B., Reina, G.A., Martin, J., Pati, S., Kotrotsou, A., Milchenko, M., Xu, W., Marcus, D., Colen, R.R., et al.: Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. Scientific reports (2020)
25. Sun, Y., Li, Z., Li, Y., Ding, B.: Improving lora in privacy-preserving federated learning. arXiv preprint arXiv:2403.12313 (2024)
26. Tan, Y., Long, G., Liu, L., Zhou, T., Lu, Q., Jiang, J., Zhang, C.: Fedproto: Federated prototype learning across heterogeneous clients. In: AAAI (2022)
27. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: Caltech-ucsd birds-200-2011. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011)
28. Wu, X., Huang, S., Wei, F.: Mixture of lora experts. In: Int. Conf. Learn. Represent. (2024)
29. Yadav, P., Tam, D., Choshen, L., Raffel, C.A., Bansal, M.: Ties-merging: Resolving interference when merging models. Adv. Neural Inform. Process. Syst. (2024)
30. Yang, E., Wang, Z., Shen, L., Liu, S., Guo, G., Wang, X., Tao, D.: Adamerging: Adaptive model merging for multi-task learning. In: Int. Conf. Learn. Represent. (2024)
31. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: International conference on machine learning. PMLR (2017)
32. Zhang, G., Wang, L., Kang, G., Chen, L., Wei, Y.: Slca: Slow learner with classifier alignment for continual learning on a pre-trained model. In: Int. Conf. Comput. Vis. (2023)

A Details on $G_{SHAP}(\omega^{l_A}, \omega^{l_B})$

Table A: Calibrated values for $G_{SHAP}(\omega^{l_A}, \omega^{l_B})$ based on ω^{l_A} and ω^{l_B} quantiles.

	$Q_B < 60$	$60 \leq Q_B \leq 80$	$Q_B > 80$
$Q_A < 25$	0.038	0.100	0.193
$25 \leq Q_A \leq 50$	0.038	0.089	0.118
$Q_A > 50$	0.038	0.078	0.093

The merging coefficients α_m^l, detailed in the main paper, are derived through a discretization procedure applied to the importance signals $\omega_m^{l_A}$ and $\omega_m^{l_B}$. To mitigate overfitting on the proxy dataset, a discretization of these values is performed instead of directly utilizing the SHAP-explained regression model's predictions.

Specifically, these values are obtained by summing a SHAP base value of 0.063 with the SHAP values of individual features, the latter being approximated through analysis of the SHAP proxy dataset figure presented in the main paper.

At the conclusion of each round, the server calculates the quantile values Q_A and Q_B for the complete distribution of $\omega_m^{l_A}$ and $\omega_m^{l_B}$, respectively, across all clients m and layers l. Subsequently, for each layer, the corresponding value is assigned based on the intervals specified in Table A, and a normalization is applied across each layer to ensure that the sum of the resulting SHAP values after the discretization of $\omega_m^{l_A}$ and $\omega_m^{l_B}$ equals 1. This procedure defines the function $G_{SHAP}(\cdot)$, which is executed server-side at the end of each communication round, returning α^l coefficients.

Conceptually, this approach rewards clients exhibiting high ω^{l_B} and low ω^{l_A} values, assigns moderate weights to clients with either high levels of both ω^{l_B} and ω^{l_A} or intermediate ω^{l_B} levels, and assigns minimal weights to clients with low ω^{l_B} values.

B Robustness in the Choice of $\omega_m^{l_A}$ and $\omega_m^{l_B}$

The selection of $\omega_m^{l_A}$ and $\omega_m^{l_B}$ as predictors of the optimal weighting coefficients λ_m^l stems from the intuition that a layer's contribution to loss variation can effectively capture its importance during server-side aggregation. We further investigate whether the number of examples seen during training could provide additional useful information for predicting the optimal coefficients.

Specifically, we initially trained the Random Forest using three features: $\omega_m^{l_A}$, $\omega_m^{l_B}$, and the number of training samples for each client. When evaluating the

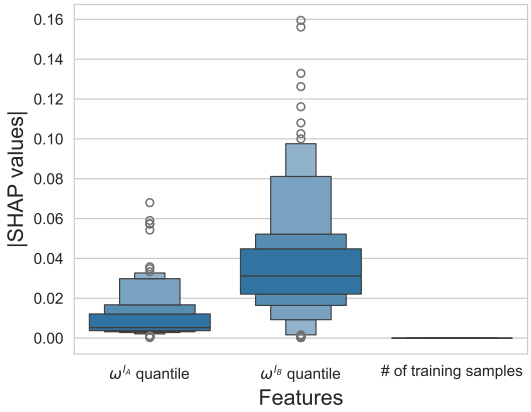

Fig. A: Evaluation of efficacy of the number of training samples in λ_m^l prediction.

SHAP values of these three features, it became evident that the impact of the number of samples is negligible, as shown in Figure A. This result is directly attributable to the fact that λ_m^l coefficients are estimated layer-wise. More importantly, this finding supports our hypothesis that $\omega_m^{l_A}$ and $\omega_m^{l_B}$ are sufficient and appropriate choices for predicting optimal merging coefficients.

C Additional a posteriori Analysis of SHAP Results

As demonstrated in the main paper, the SHAP trend observed for the ω^{l_A} and ω^{l_B} quantiles is consistent between the Tiny-ImageNet proxy dataset and other distributions. Figures Fig. B, Fig. C, and Fig. D further illustrate that this trend is consistently maintained across the CIFAR-100, EuroSAT, and CUB-200 datasets. Specifically, for CIFAR-100, the negative impact of ω^{l_A} at high quantiles is more pronounced at the upper extreme of the distribution. The impact of ω^{l_B} is generally similar across the datasets examined, with the notable exception of CUB-200, where its relative impact is lower in magnitude. Consequently, the non-decreasing trend for ω^{l_B} is not fully maintained across the entire distribution for this dataset. This observation is likely attributable to the dissimilarity between the CUB-200 dataset and the pretraining data, suggesting that the change in impact layer direction, represented by the A matrices, exerts a more significant influence on the optimal merging strategy in this case.

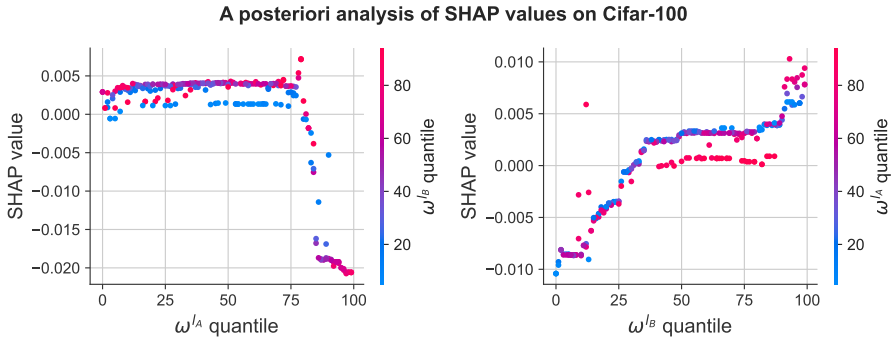

Fig. B: A posteriori analysis on $\omega_m^{l_A}$ and $\omega_m^{l_B}$ impact in predicting λ_m^l coefficients on CIFAR-100.

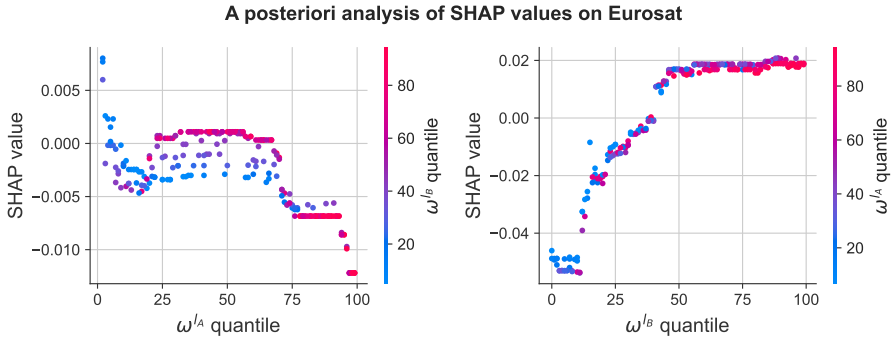

Fig. C: A posteriori analysis on $\omega_m^{l_A}$ and $\omega_m^{l_B}$ impact in predicting λ_m^l coefficients on EuroSAT.

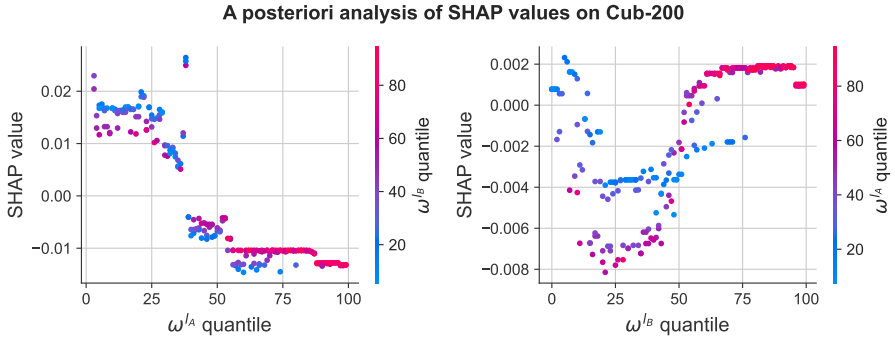

Fig. D: A posteriori analysis on $\omega_m^{l_A}$ and $\omega_m^{l_B}$ impact in predicting λ_m^l coefficients on CUB-200.