Towards Highly Heterogeneous Federated Learning: A Graph-Based Framework for Architectural and Data Heterogeneity

 Furkan Pala and  Islem Rekik *

BASIRA Lab, Imperial-X (I-X) and Department of Computing, Imperial College London, London, United Kingdom

Abstract. Hospitals train more accurate diagnostic backbones by collaborating, but privacy rules forbid sharing patient images, and federated learning (FL) assumes every hospital trains the same backbone. Real hospitals do not: a small clinic runs a convolutional backbone while a research hospital runs a transformer, and the weights of different backbones cannot be averaged. Recent FL studies address this architectural heterogeneity (AH), but the task is harder under data heterogeneity (DH), where clients hold data that is not independent and identically distributed (non-IID). We propose graph-based federated learning (GFL): each client keeps its data and its backbone private and federates only one small graph neural network (GNN), optimized over the graph representations of the backbones. Tested across both AH and DH against benchmarks for clients with different backbones, GFL is the most accurate on three of the four most architecturally varied backbone combinations, and our theoretical analysis predicts when federating helps. Our code is available at https://github.com/basiralab/GFL. [1]

Keywords: Federated learning · Architectural heterogeneity · Data heterogeneity · Medical image classification · Graph neural networks.

1 Introduction

Hospitals build better diagnostic backbones by collaborating, but privacy regulation and institutional ownership of patient records keep images inside the site that collected them. Federated learning (FL) was built for this: each hospital, called a client, trains on its own data and sends only backbone updates to a server, which averages them position by position into a shared backbone [1,2,3].

* Corresponding author: f.pala23@imperial.ac.uk,i.rekik@imperial.ac.uk, http://basira-lab.com, GitHub: https://github.com/basiralab/GFL

[1] This paper has been accepted for a **Poster Presentation** at the DeCaF MICCAI 2026 workshop.

This requires every client to train the same backbone [1], and clinical deployments break the requirement on two axes. Case mix, scanners and referral patterns differ between hospitals, so clients hold data that is not independent and identically distributed (non-IID), and weight averaging degrades as this data heterogeneity (DH) grows [4,5,6,7]. Compute differs as much as data: a small clinic runs a compact convolutional backbone while a research hospital fine-tunes a transformer, and averaging is undefined between backbones that do not share layers [8,9]. This second axis is harder, because it yields not degraded accuracy but an operation that cannot be performed at all, forcing a consortium to standardize on one backbone before it can collaborate. We call the set of backbones the clients run a backbone combination, and how far they differ architectural heterogeneity (AH).

Unified learning addresses the second axis [10]: each backbone is represented as a graph of its layers and connections, and a graph neural network (GNN) [11] reads that graph and produces parameters that scale and shift the backbone's hidden features [12]. Because the GNN reads an architecture instead of being fixed to one, a single shared GNN serves a convolutional backbone, a transformer, or a fully-connected backbone alike. That framework, however, pools all data on one server, which the privacy rules that motivate FL do not permit. We propose graph-based federated learning (GFL), which moves the shared GNN into federated training: each client keeps its data and its backbone private and shares only the small GNN, which the server averages each round. To our knowledge this is the *first FL framework whose shared component reads each client's architecture*.

We compare three learning paradigms that differ by exactly one component: each client training its backbone alone (Individual); the GNN optimized at each client but kept private (Graph-Local, GL); and the GNN federated (GFL). *Hypothesis 1*: federating the GNN raises accuracy above training alone at most clients, and the size of that gain follows a break-even between the variance a client removes by federating and the bias its own DH adds. *Hypothesis 2*: the difficulty the shared GNN faces is measured directly from the spread of the gradients clients contribute to it, which grows with AH. We confirm both (Secs. 4.1–4.3).

2 Related Work

Federating under DH. FedProx [6] and SCAFFOLD [7] correct the drift skewed labels induce, and Dirichlet partitioning is the standard way to dial DH up and down [5]. All keep the backbone fixed, so they never face the harder axis.

Federating under AH. HeteroFL [8] lets clients train width-pruned slices of one shared template, so the backbones still share a common structure, and distillation methods share soft predictions on a public dataset [9]. The closest line to ours shares small, architecture-independent summaries instead of weights: FedProto shares one average feature vector per class [13], FedGH a single classification head that every client trains on its own features [14], and FedTGP

Table 1. What each method must expose and require. A check mark is the desirable property.

	Privacy: no class info shared	No common feature space	Server only averages	Label-set independent	Robust to missing classes
FedProto [13]	×	×	✓	×	×
FedGH [14]	×	×	×	×	×
FedTGP [15]	×	×	×	×	×
GFL (ours)	✓	✓	✓	✓	✓

trainable class prototypes refined with a contrastive objective [15]. We compare against all three. They differ from ours on four counts (Table 1): every message is tagged with its class, so the server learns each client's label distribution and class-average features invite reconstruction attacks; all clients must agree on one common feature space; FedGH and FedTGP optimize on the server every round rather than averaging, so it must be trusted and must own an accelerator; and a client with no samples of a class has nothing to send, so a rare-disease site contributes least where it needs most. Our framework avoids all four, because it federates one small task-agnostic module carrying no class or label information.

Unified learning, and the gap we close. The GNN we federate is grounded in the unified learning framework (uGNN) of [10], which models each backbone as a graph and uses a shared GNN to modulate hidden features across heterogeneous architectures. However, the original uGNN requires simultaneous access to all client data, rendering its training centralized. To bridge this gap, we replace centralized joint optimization with federated averaging, transitioning uGNN into a privacy-preserving federated paradigm.

3 Methodology

3.1 Data and Learning Paradigms

Each client k owns a backbone f_{θ_k}, differing across clients in family and size. One shared three-layer graph attention network [11] g_ϕ (≈88k parameters) reads a graph G_k whose nodes are the adjustable blocks of f_{θ_k} and outputs, per block, a scale γ and a shift β. Each node carries a 20-dimensional feature vector that identifies the block it stands for. The vector records the layer type, the parameter count, the input and output widths, the kernel size, the depth of the block within the backbone, and the current statistics of its weights and activations. Two blocks of the same layer type but different capacity therefore receive different node features, and the GNN assigns them different (γ, β); in every backbone we use, all node vectors are pairwise distinct (supplementary Sec. S2). A private adapter applies them to the block's hidden state as $h \leftarrow h + \alpha_k(\gamma \odot h + \beta)$, a feature-wise linear adjustment [12] (Fig. 1b). We call this operation the feature adjustment. The learned gate α_k sets how strongly it is applied and starts at zero, so the client begins from its unadjusted backbone. Only g_ϕ is shared and averaged; θ_k, the adapter and α_k are optimized locally and never leave the client.

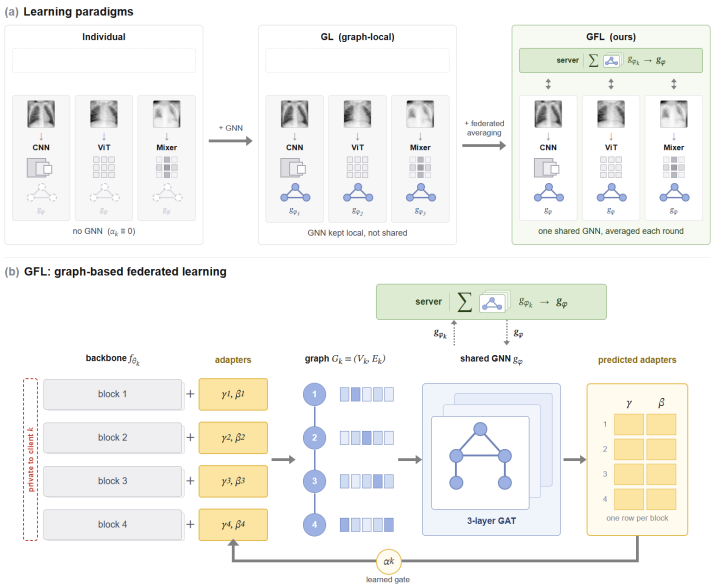

Fig. 1. Our framework. (a) The three learning paradigms differ by one component at a time: Individual trains each backbone alone, GL adds the GNN but keeps it at the client, and GFL federates it. **(b)** How GFL works, at one client and the server. G_k is drawn with four nodes for legibility; real graphs have one node per adjustable block, from 10 (SimpleCNN) to 87 (ViT-Tiny) (supplementary Sec. S1).

Our three learning paradigms change exactly one component at a time (Fig. 1a). **Individual** turns the GNN off ($\alpha_k \equiv 0$). **GL** optimizes the GNN at each client but never shares it, showing what a locally optimized GNN is worth on its own. **GFL** adds federated averaging, weighted by client sample count, showing the value of federating.

We train every method under one identical protocol, on six backbones spanning four inductive-bias types and three medical imaging datasets at 28×28, all from scratch (Table 2). The three benchmarks of Sec. 2 require one common embedding space, so we give every backbone a projection layer to 128 dimensions; our GNN needs none. We control DH with a Dirichlet split of concentration α, where lower α makes each client's labels more skewed, and AH by which backbone families the clients run. We measure AH as the mean pairwise Jensen–Shannon divergence (JSD, base 2) between the clients' layer-type distributions, $\mathrm{AH} = \binom{m}{2}^{-1} \sum_{i<j} \mathrm{JSD}(p_i \,\|\, p_j)$, where m is the number of clients and p_i is the histogram of client i's blocks over the twelve-entry layer-type vocabulary from which its graph is built. The metric lies in $[0, 1]$, is 0 when every client runs the same backbone, and can be computed from the model definitions alone, before any training takes place; supplementary Sec. S3 reports the histograms and the pairwise divergences, and states the cases it cannot distinguish. We chose the six combinations to walk this axis in roughly even steps while adding one

Table 2. Backbones (trained from scratch) and datasets. Parenthesized names are the short forms used in Table 3; MLP is a multilayer perceptron. MIMIC-CXR is labeled with CheXbert [16].

Backbone	Type	Params
SimpleCNN (CNN)	convolutional	0.37M
VGG9 [17]	convolutional	4.82M
ResNet18 [18]	convolutional	11.17M
ViT-Tiny (ViT) [19]	attention	5.36M
MLP-Mixer-S (Mixer-S) [20]	mixing	1.85M
MLP	fully-connected	2.71M

Dataset	Modality	Task
PathMNIST [21,22]	histopathology	9-class
HAM10000 [23]	dermatoscopy	malignant/benign
MIMIC-CXR [24]	chest X-ray	abnormal/normal

inductive-bias family at a time, so that AH, and not any single architecture, is what changes from row to row in Table 3. Every run is repeated with three seeds, the same three for every method, and we report the mean ± standard deviation (sd). A centralized model pools all client data and so breaks the privacy constraint every method here respects; we report it only as an upper bound.

3.2 Theoretical Analysis

We identify the two factors that make federating the GNN difficult. Let $\phi^\star$ be the fixed point of the federated averaging of g_ϕ and $\phi_k^\star$ what client k would converge to alone. Because α_k starts at zero, the client's gradient factorizes as $\nabla_\phi F_k = \alpha_k G_k$, so at initialization GFL computes exactly what Individual training computes and cannot change the client's accuracy. This guarantee holds at initialization only. Once the client raises α_k, the shared gradient begins to act on its backbone, and nothing in the framework prevents that client from finishing below where it would have finished on its own; under the strongest label skew we test, it does exactly that (Sec. 4.2). The gate governs how much shared signal a client admits, but it does not certify that admitting the signal will help. This makes GFL a partial-personalization model [25], whose $\mathcal{O}(1/\sqrt{T})$ guarantee we instantiate for g_ϕ, with T the number of federated rounds.

The distance between the shared solution and a client's own best solution splits in two, $\|\phi^\star - \phi_k^\star\| \le 2\,\zeta_{\text{data}} + \zeta_{\text{arch}}$, where ζ_{data} measures how much the clients' gradients spread apart because their data differs and ζ_{arch} because their architectures do. The terms are separable by design: sweeping α moves ζ_{data} at fixed backbones, and changing which backbone families the clients run, at comparable data, moves ζ_{arch}. Training trajectories confound ζ_{arch} with the optimizer's normalization, so we measure it from raw gradients spread on a common data mixture, at a shared ϕ. Gradients shrink as training converges, so we take that spread relative to their own size, $\zeta_{\text{arch}}/\|\bar{g}\|$, which is 1 when clients pull in mutually independent directions and below 1 when they share a direction worth averaging.

Recasting the participation threshold of [26] gives the prediction we test. Under a quadratic surrogate, a client's own estimate of ϕ is unbiased and has variance σ^2/n_k, while the federated average has the smaller variance σ^2/n but acquires a bias b_k because it is fitted to the other clients' data as well. Client k therefore gains from federating when $\sigma^2(1/n_k - 1/n) - \|b_k\|^2$ is positive, with n_k

Table 3. Backbone combinations with growing AH on PathMNIST. Each row lists the backbone every client runs (short names, Table 2); Fam. is the number of distinct families. Clients hold comparable data. Pooled-test accuracy at the validation-selected round, mean±sd over three seeds, best per row in bold.

Backbone combination	Fam.	AH	Individual	FedProto	FedGH	FedTGP	GL	GFL
5 × CNN	1	0.00	0.864±.006	0.871±.004	0.870±.006	**0.888**±.003	0.852±.011	0.875±.003
4 × CNN + ResNet18	2	0.11	0.872±.006	0.879±.006	0.871±.008	**0.886**±.003	0.863±.011	0.879±.006
4 × CNN + Mixer-S	2	0.30	0.860±.004	0.866±.005	0.866±.003	0.865±.008	0.859±.009	**0.871**±.005
CNN + MLP + 3 × Mixer-S	3	0.36	0.790±.004	0.788±.009	0.789±.005	0.796±.002	0.794±.009	**0.797**±.004
CNN + 2 × VGG9 + 2 × ViT	3	0.48	0.849±.005	0.845±.005	0.854±.013	**0.861**±.004	0.834±.016	0.859±.006
CNN + 2 × ResNet18 + ViT + MLP	4	0.60	0.815±.001	0.806±.003	0.814±.004	0.818±.004	0.811±.005	**0.820**±.004

client k's sample count, n the total and σ^2 the per-sample variance of a client's estimate. Federating helps when the variance a client removes outweighs the bias its own heterogeneity adds. The first term does not depend on DH while the second grows with it, so the benefit is largest at comparable data and shrinks as distributions diverge, crossing zero at a break-even point that Sec. 4.2 locates. Supplementary Sec. S4 derives both results in full. Supplementary Sec. S6 reports our attempt to estimate σ^2 and $\|b_k\|$ separately from the runs themselves, which three seeds do not resolve, so we test the break-even condition through the ordering it predicts rather than through its two terms taken apart.

4 Experiments & Results

4.1 Federating the GNN Improves Every Backbone Combination

We built six backbone combinations spanning AH from identical backbones (AH=0.00) to four families (AH=0.60) (Table 3), and write Δ for GFL accuracy minus Individual accuracy on the same split. **We found that federating the GNN raises accuracy above training alone on all six combinations**, Δ running from +0.005 to +0.011, and in 16 of the 18 (combination, seed) runs; it also raises accuracy at every client count from two to six.

The gain is not an artifact of averaging over clients, because it reaches most clients individually. Resolved to the client level, federating the GNN raises accuracy at 25 of the 30 clients and in 63 of the 90 (client, seed) pairs. The five clients that do not improve are spread over four combinations rather than concentrated in the most architecturally varied ones (supplementary Sec. S5).

Δ comes from federating the GNN, not from optimizing it locally. The GL column isolates this: a privately kept GNN beats Individual training on one of the six combinations here and on three of the twelve comparable-data comparisons overall, whereas federating the same GNN wins eleven of those twelve.

Against the three benchmarks, our framework is the most accurate on three of the four most architecturally varied combinations, which is the regime we target. FedTGP [15] leads on the two combinations whose backbones are identical or nearly so, and it does so while exchanging class-linked prototypes and

running contrastive optimization on the server every round, neither of which our framework requires (Table 1).

4.2 Locating the Break-Even Point

Hypothesis 1 predicts that Δ shrinks as client distributions diverge and crosses zero. We tested this on a grid crossing three backbone combinations, AH=0.00 to AH=0.60, with three levels of DH (Fig. 2, left). **We found the predicted crossing, and we found it at every AH level:** Δ rises as client distributions come into agreement in all three combinations, turning positive at comparable data and negative under the strongest skew. Replicating one ordering at three backbone combinations is direct evidence that the decomposition of Sec. 3.2 names the right two factors and that they act separately, and the sign change survives unequal client data: across a five-level DH sweep, Δ is -0.015 under the strongest skew and returns to parity as the skew weakens. The number of clients that gain moves with the mean, rising from 1, 0 and 1 of five under the strongest skew to 5, 3 and 3 of five at comparable data, at AH=0.00, 0.36 and 0.60. This gives a consortium a decision rule: federate the GNN when member label distributions are broadly comparable, and optimize it privately when they are extremely skewed, which costs nothing, since GL then beats Individual training on all three combinations.

4.3 Measuring What Limits Federating

We applied the gradient probe of Sec. 3.2 to the same six combinations (Fig. 2, right), which holds the data fixed so that architecture is the only thing that varies, and tests *Hypothesis 2*. **We discovered that the architecture term of our decomposition is measurable directly from gradients, and that it grows with AH:** ζ_{arch} doubles from identical backbones to the most varied combination, and its scale-free form $\zeta_{\text{arch}}/\|\bar{g}\|$ rises from 0.90 to 1.15 (ρ=+0.83). Alike backbones pull in a common direction that averaging reinforces, unalike ones partly cancel, and that caps how much a shared GNN carries. The one combination reading high against its AH is also the one whose gradients are smallest, which inflates the ratio. We report this relationship as a trend and not as a tested law: six points are too few for the correlation to reach significance, no coefficient we compute attains $p < 0.05$ under an exact permutation test, and supplementary Sec. S7 lists every coefficient together with its p-value.

4.4 Three Medical Imaging Types

Finally we tested whether the break-even prediction transfers beyond histopathology, comparing the three learning paradigms on all three imaging types (supplementary Table S12), each at the loss its label balance calls for [27] and at the DH level its label structure allows. **We found that our framework behaves on dermatoscopy and chest radiography exactly as the decomposition predicts for that regime.** All three operating points sit at or below

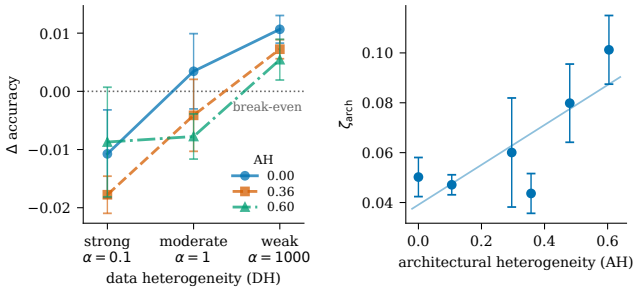

Fig. 2. Left: the break-even point. Δ against DH, set by the Dirichlet concentration α; one line per backbone combination, labeled by its AH. **Right: what limits federating.** ζ_{arch} over the six combinations of Table 3, with a least-squares fit. Error bars are the sd over three seeds.

the break-even point of Sec. 4.2, and at all three our framework tracks Individual training within one sd, and it does so on two imaging types the decomposition was never fitted to.

5 Discussion & Conclusion

Accuracy alone does not decide whether a consortium can adopt a method; what the method demands of each member does, and ours demands little. Each site keeps the backbone it already runs, so joining needs no common architecture and no new hardware; the server averages an 88k-parameter module, so it needs no accelerator and no trust beyond arithmetic; and because the gate starts at zero, each site controls how much shared signal it admits. A centralized model leads on every imaging type, and the gap is not a limit of the small backbones, since pooling lifts the CNN to within half a point of ResNet18, thirty times its size. The bottleneck is each client's feature extractor, which federating a GNN does not enlarge, so future work belongs there. Across the full campaign our framework raises accuracy over training alone in 14 of the 22 comparisons and in 11 of the 12 at comparable data, ahead of FedProto [13] and FedGH [14]. Three limitations bound our claims. The effect sizes are small, because Δ is under one accuracy point and lies within a seed's standard deviation, so our evidence rests on the consistency of the effect rather than on any single margin. The images are 28×28, so results may differ at clinical resolution. And we study one GNN, so our findings may not transfer to every heterogeneous-federation method. We federated a shared GNN previously restricted to centralized training, so hospitals with different backbones can collaborate without sharing either. Accuracy rises on every backbone combination, client count and most clients; our decomposition predicts a break-even point we locate and confirm on two further imaging types (*Hypothesis 1*); and gradient spread grows with AH (*Hypothesis 2*).

References

1. McMahan, B., Moore, E., Ramage, D., Hampson, S., Aguera y Arcas, B.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS). pp. 1273–1282 (2017)
2. Rieke, N., Hancox, J., Li, W., Milletarì, F., Roth, H.R., Albarqouni, S., Bakas, S., Galtier, M.N., Landman, B.A., Maier-Hein, K., et al.: The future of digital health with federated learning. npj Digital Medicine **3**(1), 119 (2020)
3. Sheller, M.J., Edwards, B., Reina, G.A., Martin, J., Pati, S., Kotrotsou, A., Milchenko, M., Xu, W., Marcus, D., Colen, R.R., Bakas, S.: Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. Scientific Reports **10**(1), 12598 (2020)
4. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-IID data. arXiv preprint arXiv:1806.00582 (2018)
5. Hsu, T.M.H., Qi, H., Brown, M.: Measuring the effects of non-identical data distribution for federated visual classification. arXiv preprint arXiv:1909.06335 (2019)
6. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. In: Proceedings of Machine Learning and Systems (MLSys). vol. 2, pp. 429–450 (2020)
7. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S.J., Stich, S.U., Suresh, A.T.: SCAFFOLD: Stochastic controlled averaging for federated learning. In: Proceedings of the 37th International Conference on Machine Learning (ICML). pp. 5132–5143 (2020)
8. Diao, E., Ding, J., Tarokh, V.: HeteroFL: Computation and communication efficient federated learning for heterogeneous clients. In: International Conference on Learning Representations (ICLR) (2021)
9. Lin, T., Kong, L., Stich, S.U., Jaggi, M.: Ensemble distillation for robust model fusion in federated learning. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 33, pp. 2351–2363 (2020)
10. Pala, F., Rekik, I.: GNN-based unified deep learning. In: Predictive Intelligence in Medicine: 8th International Workshop, PRIME 2025, Held in Conjunction with MICCAI 2025. pp. 34–46. Lecture Notes in Computer Science, Springer, Cham (2025). https://doi.org/10.1007/978-3-032-07904-6_4
11. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (ICLR) (2018)
12. Perez, E., Strub, F., de Vries, H., Dumoulin, V., Courville, A.: FiLM: Visual reasoning with a general conditioning layer. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
13. Tan, Y., Long, G., Liu, L., Zhou, T., Lu, Q., Jiang, J., Zhang, C.: FedProto: Federated prototype learning across heterogeneous clients. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 8432–8440 (2022)
14. Yi, L., Wang, G., Liu, X., Shi, Z., Yu, H.: FedGH: Heterogeneous federated learning with generalized global header. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 8686–8696 (2023)
15. Zhang, J., Liu, Y., Hua, Y., Cao, J.: FedTGP: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 16768–16776 (2024)

16. Smit, A., Jain, S., Rajpurkar, P., Pareek, A., Ng, A.Y., Lungren, M.: CheXbert: Combining automatic labelers and expert annotations for accurate radiology report labeling using BERT. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1500–1519 (2020)
17. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR) (2015)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
19. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (ICLR) (2021)
20. Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., Lucic, M., Dosovitskiy, A.: MLP-Mixer: An all-MLP architecture for vision. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 34, pp. 24261–24272 (2021)
21. Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., Ni, B.: MedMNIST v2 – a large-scale lightweight benchmark for 2D and 3D biomedical image classification. Scientific Data **10**(1), 41 (2023)
22. Kather, J.N., Krisam, J., Charoentong, P., Luedde, T., Herpel, E., Weis, C.A., Gaiser, T., Marx, A., Valous, N.A., Ferber, D., et al.: Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. PLoS Medicine **16**(1), e1002730 (2019)
23. Tschandl, P., Rosendahl, C., Kittler, H.: The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Scientific Data **5**(1), 180161 (2018)
24. Johnson, A.E.W., Pollard, T.J., Berkowitz, S.J., Greenbaum, N.R., Lungren, M.P., Deng, C.Y., Mark, R.G., Horng, S.: MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports. Scientific Data **6**(1), 317 (2019)
25. Pillutla, K., Malik, K., Mohamed, A.R., Rabbat, M., Sanjabi, M., Xiao, L.: Federated learning with partial model personalization. In: Proceedings of the 39th International Conference on Machine Learning (ICML). vol. 162, pp. 17716–17758 (2022)
26. Donahue, K., Kleinberg, J.: Model-sharing games: Analyzing federated learning under voluntary participation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 5303–5311 (2021)
27. Ren, J., Yu, C., Sheng, S., Ma, X., Zhao, H., Yi, S., Li, H.: Balanced meta-softmax for long-tailed visual recognition. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 33, pp. 4175–4186 (2020)